DTIC FILE COPY

AD-A218 287

FROM: AFIT/CI

31 Jan 90

SUBJ: Review of Thesis/Disseration for Public Release

TO: PA

Request you review the attached for public release prior to being sent to DTIC.

ERNEST A. HAYWOOD, 1st Lt, USAF
Executive Officer
Civilian Institution Programs

1 Atch
Thesis
~~Disseration~~
by Camejo

1st Ind, AFIT/PA

8 FEB 1990

TO: CI

Approved/~~Disapproved~~ for public release.
Log Number: 89-178

Harriet D Moultrie

HARRIET D. MOULTRIE, Capt, USAF
Director, Office of Public Affairs

PA Log Number 90-02-09

90 ... 40

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFIT/CI/CIA-89-178
6a. NAME OF PERFORMING ORGANIZATION AFIT STUDENT AT PURDUE UNIV; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFIT/CIA
6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB OH 45433-6583
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) (UNCLASSIFIED)

AN EXPERT SYSTEM FOR THE DESIGN OF HEATING, VENTILATING, AND AIR-CONDITIONING SYSTEMS

12. PERSONAL AUTHOR(S)
PEDRO JOSE CAMEJO

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
THESIS/DISSERTATION	FROM ______ TO ______		

16. SUPPLEMENTARY NOTATION
APPROVED FOR PUBLIC RELEASE IAW AFR 190-1
ERNEST A. HAYGOOD, 1st Lt, USAF
Executive Officer, Civilian Institution Programs

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL ERNEST A. HAYGOOD, 1st Lt, USAF	22b. TELEPHONE (Include Area Code) (513) 255-2259; 22c. OFFICE SYMBOL AFIT/CI

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

AFIT/CI "OVERPRINT"

PURDUE UNIVERSITY

Graduate School

This is to certify that the thesis prepared

By Pedro Jose Camejo

Entitled An Expert System For The Design Of Heating, Ventilating, And Air-Conditioning Systems

Complies with University regulations and meets the standards of the Graduate School for originality and quality

For the degree of Master of Science in Mechanical Engineering

Signed by the final examining committee:

Douglas C. Hittle, chair
D. C. Hittle

D. R. Tree
D. R. Tree

R. J. Bernhard

Approved by the head of school or department:

December 14, 19 89 F. P. Incropera/M. R. L'Ecuyer
M. R. L'Ecuyer
School of Mechanical Engineering

This thesis ☐ is ☒ is not to be regarded as confidential

Douglas C. Hittle
Major professor

Grad. School
Form No. 9
Revised 9-85

AN EXPERT SYSTEM FOR THE DESIGN OF HEATING, VENTILATING, AND AIR-CONDITIONING SYSTEMS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Pedro Jose Camejo

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

December 1989

ACKNOWLEDGMENTS

I would like to thank Professor Douglas C. Hittle, my major professor, for his guidance and support throughout my graduate work. I would also like to thank the United States Air Force for providing me with the time and financial support needed to complete this work.

Additional thanks goes to the staff of the Ray W. Herrick Laboratories for their support in general, and for providing the software used in this project.

Special appreciation and thanks are due to my wife, Diane, for her unceasing support and encouragement without which this project would not have been possible and for her careful and insightful editing of the thesis manuscript.

Pedro J. Camejo

Purdue University
December 1989

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Camejo, Pedro Jose. MSME., Purdue University, December 1989. An Expert System For The Design Of Heating, Ventilating, and Air-Conditioning Systems. Major Professor: Douglas C. Hittle, School of Mechanical Engineering.

Expert systems are computer programs that seek to mimic human reason. An expert system shell, a software program commonly used for developing expert systems in a relatively short time, was used to develop a prototypical expert system for the design of heating, ventilating, and air-conditioning (HVAC) systems in buildings. Because HVAC design involves several related knowledge domains, developing an expert system for HVAC design requires the integration of several smaller expert systems known as knowledge bases. The expert system shell has been used to develop several HVAC design knowledge bases. A menu program and several auxiliary programs for gathering data, completing calculations, printing project reports, and passing data between the knowledge bases are needed and have been developed to join the separate knowledge bases into one simple-to-use program unit. EMK

Keywords: CYBERNETICS, EXPERT SYSTEMS, HVAC, AIR CONDITIONING EQUIPMENT, Heating, Ventilation. EMK

CHAPTER I

INTRODUCTION

Introduction to Expert Systems

Knowledge-based expert systems are computer programs that use heuristics (rules-of-thumb) to solve problems in a given knowledge domain. Paraphrasing Mike Van Horn, several characteristics that distinguish expert systems from conventional software programs are as follow (Van Horn 1986):

> 1. An expert system does not require all decision rules in the program and all data used to solve the problem to be reduced to numbers and algebraic equations.
>
> 2. An expert system allows more than one solution to be computed for any one set of data.
>
> 3. An expert system program is capable of providing default data or of otherwise continuing until a solution is reached even if the user does not have all the needed data. Thus, missing data do not halt program execution.
>
> 4. An expert system program assigns a certainty number to the solution or solutions it computes. For example, if much input data is missing the expert system will provide a solution with a low certainty level.

Knowledge-based expert systems form a branch of artificial intelligence (AI) that has been under study since the early 1960's. Peter Jackson (Jackson 1986) states that AI is the part of computer science concerned with systems that "exhibit the characteristics associated with intelligence in human behavior: understanding, language, learning, reasoning, solving problems, and so on." Some experts no longer consider knowledge-based expert systems a branch of AI as they view the expert system as incapable of dealing with completely new conditions; i.e. conditions that have not been programmed

into the expert system. Nevertheless, knowledge-based expert systems are the only branch of AI to date that has had successful commercial applications.

In the past 30 years knowledge-based systems have been applied to such varied domains as medicine, genetics, chemistry, geology, economics, and civil, mechanical, and electrical engineering. The literature provides a thorough description of knowledge domains to which expert systems have been applied, including discussions on the practical success of some of the systems developed. Simons (1985), Townsend and Feucht (1986), and Van Horn (1986) all present detailed historical data.

Mechanical engineering applications of expert systems include acoustics, controls, machine design, and fluid mechanics. Some literature was also found concerning expert systems in the construction industry. Gero (1985) and Kosten and Maher (1986) both discuss construction design expert systems. More importantly, three papers presented at the 1988 Winter meeting of the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) fall under the category of expert systems in heating, ventilating and air conditioning (HVAC). These papers included an expert system for the knowledge-based interpretation and control of building energy consumption (Haberl et al. 1988), an input generating expert system (commonly known as a front end program) for a building system simulation program (Liu and Kelly 1988), and a thorough description of the knowledge acquisition process for HVAC expert systems programming (Brothers 1988).

Objective

The objective of this research was to demonstrate the feasibility of an expert system for the design of HVAC systems in buildings and to develop the structure of such an expert system. There are an abundant number of conventional HVAC design programs available to today's engineers. Listings of such available programs are found in Andrade and Degelman (1986), Daryanani (1980), Mueller and Associates (1986), and

O'Connel et al. (1985). This project is not intended to add to this list or to replace existing programs designed to do engineering calculations, but instead to investigate an alternative method of using computers in HVAC design.

The difference between the existing programs and an HVAC design expert system is revealed in the basic definition of an expert system. An expert system is a computer program which mimics a human expert in a given knowledge domain. The expert system asks questions to obtain pertinent data, uses conventional software programs to calculate other data, and mimics reason to reach the best solutions to a problem. In doing so, the expert system itself can utilize several of the already available conventional software programs to complete HVAC designs.

Ultimately the final justification for this project and the final test for its worth will be the usefulness of the final product. At this stage of program development, the worth of the project must be measured by projected benefits. Paraphrasing Van Horn, the following are three benefits of expert systems (Van Horn 1986):

> 1. The best expertise in the field is made available to many people. If the expert system is used as a learning tool, many can learn what the masters know.
>
> 2. Expert systems allow experts to handle even more complex problems rapidly and reliably.
>
> 3. Expert systems are very thorough and systematic; no factors are forgotten.

These benefits can be related to HVAC design and the building industry. The main benefit of this expert system will be to guide and teach inexperienced HVAC designers. Many inexperienced engineers are placed in positions where they are expected to design complex HVAC systems. A common comment from these engineers is that while design manuals provide up to 80% of the required knowledge, the engineer is left to make up the rest by whatever means are available. System and controls selection and system constructibility and maintainability are some of the areas that suffer most in the designs completed by novice engineers.

The other benefits of this expert system will be to provide the experienced designer with a means to both increase his productivity and check his design against a computer's unfailing memory. The knowledge-based expert system does not replace the expert; rather, it is a tool to be used by the expert. The constant pressure placed on designers to produce designs at the lowest possible cost can often lead to faulty design. With the ever increasing need to produce energy efficient HVAC systems the expert system stands as a possible solution to the problem of producing optimal designs in minimum time. It is in the often under staffed small design firm that this tool is most needed.

Recognizing that due to time constraints a complete system could not be the goal of this project, the goals set for this phase of the expert system development are as follow:

1. Develop a flexible specification that will guide the development of the expert system by providing continuity between researchers (see Appendix A).

2. Develop the complete working prototype structure of the expert system using a commercial expert system shell and/or programming languages suitable for an expert system.

3. Demonstrate the feasibility of the prototype expert system.

Hardware

The expert system described here was developed on a PC-AT compatible microcomputer with 512K of RAM, color graphics board, color monitor, and storage consisting of a single 20 MB Winchester drive and a single 1.2 MB floppy drive. A microcomputer was selected for this project because the intent was to develop a system for use by small architect and engineering (A&E) firms and by the design offices of the United States Air Force.

In general, the literature search (including Simons (1985), Townsend and Feucht (1986), and Van Horn (1986)) indicates that experts believe that microcomputers are too limited in their computing capability for productive expert systems. Nevertheless, this

project was developed for microcomputers because of the clear evidence that the microcomputer is the computer of choice for the building industry, whether the firm be an A & E or a building contractor. Soong (1985) empirically corroborates this last conclusion. Furthermore, the current availability of 32 BIT super micro-computers and the underlying rapid development in microcomputer technology shows that microcomputers posses or will soon posses the power needed to support large expert systems.

Scope

This initial chapter introduces artificial intelligence and expert systems, reviews the motivation for and objectives of the project, indicates the hardware choices for the project and the reasons behind those choices, and defines the scope of the project.

Chapter II, EXPERT SYSTEMS AND ARTIFICIAL INTELLIGENCE, presents the information found in the artificial intelligence (AI) literature research. The chapter concludes with a discussion of the selection process of programming software used for this project. Chapter III, STRUCTURE OF HVAC DESIGN EXPERT SYSTEM, describes the working prototype structure of the expert system and the rationale behind the structure. Included in this chapter are descriptions of the auxiliary programs that make up the expert system structure. The program listings for these programs can be found in Appendix D of Herrick Laboratory Report HL 89-37 (Camejo and Hittle 1989). Chapter IV, KNOWLEDGE PROGRAMMING, describes the process of programming knowledge with the expert system shell selected for this project. Chapter V, HVAC DESIGN KNOWLEDGE BASES explains the rationale behind the content of the knowledge bases and the methods used to encode the knowledge. Examples are included within this chapter and the complete set of rules are found in Appendixes G, H, I, J, K, and L of Herrick Laboratory Report HL 89-37 (Camejo and Hittle 1989). Chapter VI, CONCLUSIONS AND RECOMMENDATIONS, expands on the lessons learned during

the programming of the prototype expert system. These lessons should serve as valuable guides to future researchers working on similar projects.

CHAPTER II

EXPERT SYSTEMS AND ARTIFICIAL INTELLIGENCE

Characteristics of Expert Systems

General Characteristics

Knowledge-based expert systems form a branch of artificial intelligence (AI) that deals with the application of heuristic knowledge to the solution of problems in specific knowledge domains.

Several methods for the representation of knowledge, such as first-order logic, semantic network, frames, and production rules have been developed. However, a review of commercially available expert systems and expert system development tools indicates that frames and production rules are the most often used methods of knowledge representation (Townsend and Feucht 1986). Thus, knowledge-based systems are generally presented as being of only two types: "frames" systems and "production" systems. The former are commonly known as frames or object-oriented programs, while the latter are commonly known as rule-based systems.

Frames are networks organized in a hierarchical relationship with each frame containing information that applies to all frames below it (Townsend and Feucht 1986). Figure 1 shows a frames representation of a building type hierarchy. Using the jargon of frames, each rectangle is called a frame and holds information about a certain object. The information is contained in the narrow strips at the bottom of each frame. These strips are called slots. Each slot has both a name and a value. If a slot for any given frame is not

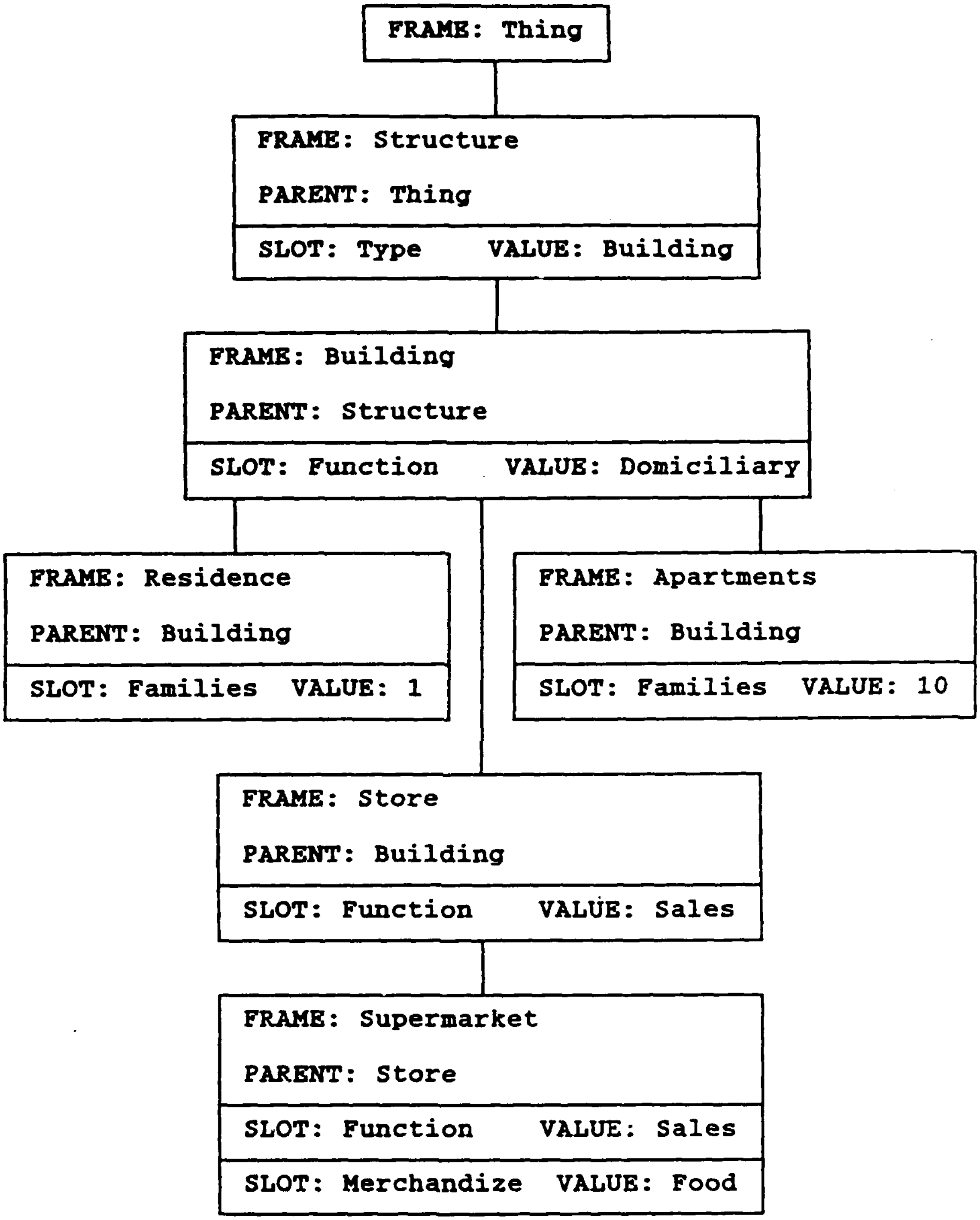

Figure 1

Partial frames representation of
building type classification.

explicitly listed and given a value, it is implicitly understood that the frame inherits the slots and values of all its ancestors.

To further illustrate the concept of frames, assume an expert system using the hierarchy of Figure 1 needs to know the value of the *function* slot for the *supermarket* frame. The expert system would access the *supermarket* frame and find the value *sales* in the *function* slot. Further assume that the expert system now needs the value of the *function* slot in the *apartments* frame. The expert system would access the *apartments*, frame and upon not finding the *function* slot would determine the *parent* of the *apartments* frame. In this case it would find that the *function* inherited by the *apartments* frame is *domiciliary*.

According to G. L. Simons (Simons 1985) the most appropriate use for frames is "to assist applications in such areas as computer vision and natural language understanding." Since the slots that make up a frame can be used to store values, procedures, or rules, it is possible to combine frames and production rules in one expert system. In fact, because each type of knowledge representation has its own best application, the more knowledge representation methods a system has the better.

In the case of the expert system described in this thesis, the knowledge representation system is production rules. Since production rules (e.g., antecedent and consequent type rules) are the identifying characteristic of rule-based systems, this expert system is properly described as a rule-based system.

Rule-based expert systems are the most common type of expert system found in the literature. While it is true that there is some disagreement among the AI experts about what makes a true expert system, the literature generally agrees with the following list of criteria for "real" expert systems. According to the list an expert system:

"1. Contains the heuristic knowledge of an expert.

2. Interfaces with the users and developers in natural language.

3. Asks questions to obtain needed data.

4. Is easily refined and upgraded without extensive reprogramming.

5. Can explain its conclusions, line of reasoning, and why it needs the input it requests.

6. Accepts uncertain inputs and assigns them certainty factors.

7. Calculates using these certainty factors to give answers with a certain level of confidence.

8. Learns from its own performance."

The more of these criteria that a system meets, the more likely the experts will agree that it is indeed an AI-based expert system.

Expert System Shell Characteristics

Early in the development of rule-based expert systems it was found that changing the knowledge domain of an existing expert system is a task that does not require changing the entire expert system program. Only the knowledge rules need to be changed. This idea evolved into what is commonly known as an expert system shell.

The structure of an expert system shell is shown in Figure 2. This structure gives the expert system the flexibility required for upgrade of the knowledge base without extensive reprogramming. One way to explain the structure shown in Figure 2 is to explain it in terms of how the expert system emulates the human expert. According to J. S. Gero (Gero 1985) the expert system "separates the expert's knowledge from the expert's behavior." The rule editor allows the system developer or developer/user to build a data base of production rules commonly known as a knowledge base. The expert's knowledge of a given domain is encoded into the knowledge base without altering the inference engine and natural language interface. The inference engine and natural language interface emulate the behavior of the expert by asking appropriate questions, explaining why it asks those questions, reaching a conclusion, explaining how it reached the conclusion, and stating its confidence in the conclusion. The separation of

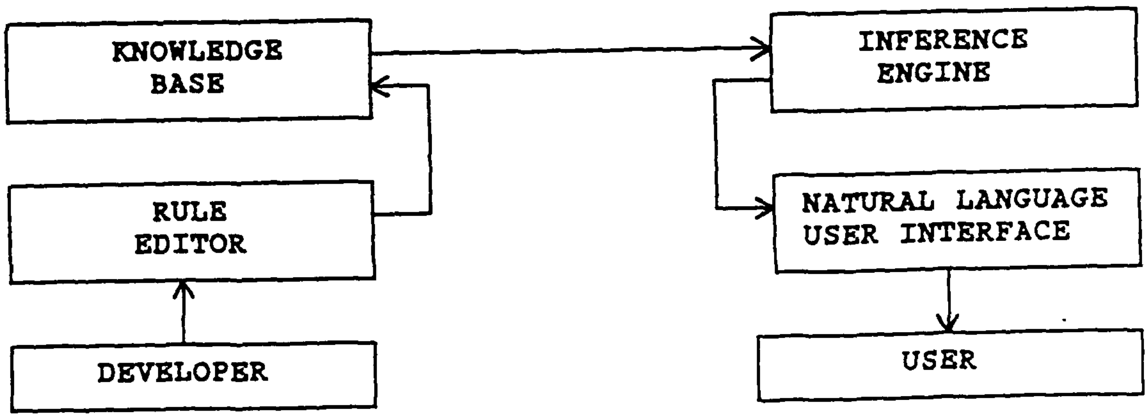

Figure 2

Structure of expert system shells.

the knowledge from the behavior allows a single inference engine and natural language interface to be used with knowledge bases in many different domains.

Programming Considerations

General

The programming considerations described here were researched from the viewpoint that all features of expert system programming were tools for the development of this HVAC design expert system. In order to do this, it was first necessary to formalize the HVAC design process so that as each programming tool was considered the design process could be reviewed to see how the particular tool might be used in the HVAC design expert system. The HVAC design process that was used for this purpose is described in section 15 of the HVAC design expert system specification under major heading HVAC DESIGN PRODUCTION SCHEDULE. This specification is found in Appendix A. In summary, the design process is one of gathering project data, completing a rule-of-thumb design to give the members of the design team and the owners an initial idea of the scope of the HVAC phase of the project, and completing the detailed design with engineering calculations, system selections, and drawings and specifications.

AI Programming Environment

The first decision to be made in the development of this expert system was the selection of the appropriate AI programming method. The two general methods considered were expert system shells and AI programming languages. As has been noted, expert system shells are generic expert systems sold without a knowledge base. The shells provide a ready programmed user interface, an inference engine, and a rule editor for building the knowledge base. Shells are written in many different languages, including commonly used languages such as BASIC, C, and Pascal. Shells are also written in LISP, PROLOG and OPS5, which are three of the better known AI languages.

LISP has existed since the 1950's and is a language that can be used much like FORTRAN for conventional programs that use mathematic algorithms. The AI power of LISP comes from its ability to perform operations on non-numeric data with the ease that FORTRAN operates on numeric data. LISP provides a simple medium by which properties and property values (numeric or non-numeric) can be assigned to an object. The frames knowledge representation method discussed earlier in this chapter is easily constructed using LISP's ability to assign and retrieve property values. For example, the following statement lines are a LISP implementation of the last frame on Figure 1:

```
->(putprop 'Supermarket 'Sales 'Function)
->(putprop 'Supermarket 'Food 'Merchandize)
```

These two lines assign the values "Sales" and "Food" to the "Function" and "Merchandize" properties of an atom called "Supermarket." The following lines show how one could retrieve the stored values from the LISP environment. The second and fourth lines represent LISP's response to the first and third statement lines:

```
->(get 'Supermarket 'Function)
->Sales
->(get 'Supermarket 'Merchandize)
->Food
```

The concepts of the parent-child object relationship and inheritance can also be easily programmed in the LISP environment, thus completing the frames implementation described in Figure 1.

One characteristic that distinguishes LISP from other AI languages is that LISP is a procedural language. PROLOG and OPS5, on the other hand, are non-procedural, or declarative languages. Versions of OPS5 are written in LISP while versions of PROLOG are written in C or assembler. Essentially PROLOG and OPS5 work with production rules and have their own built-in inference algorithm. Bratko (1986) tells us that PROLOG involves "what-type" thinking while LISP, FORTRAN and other FORTRAN-

like languages involve "how-type" thinking. PROLOG and OPS5 are languages built for rule-based expert systems.

Programming a non-trivial expert system using a programming language (as opposed to using an expert system shell) is a major task even for an experienced programmer. The use of an expert system shell, however, dramatically reduces both the development time and required programming expertise. As this project attempts to develop a prototype HVAC design expert system in minimum time, a shell was selected over programming languages.

It should be noted that the obvious advantage of AI languages over shells is flexibility. When developing an expert system using the AI languages, the developers can build the system to suit the specific knowledge domain. Shells, on the other hand, are either generic or are the result of removing the knowledge base from a custom designed expert system.

Shell Selection

Once the decision was made to use a shell for the HVAC design expert system, the next logical step was to determine which of the features found in commercial shells were advantageous for the HVAC design knowledge domain so that a suitable shell could be selected.

Since it was difficult to foresee all of the programming features that would be needed in the development of the HVAC design expert system, the first desirable attribute for the candidate shell was versatility (i.e., a system with several modes of operation). A further problem was that, except for the differences in retail prices, the available shells were difficult to differentiate. The descriptive literature for the different shells was not detailed enough to provide a good comparison between them, and even demonstration disks did not show one product to be superior to another.

Shell Structure. Before looking at the shell selection process it is advantageous to again refer to Figure 2 and describe each of the elements of an expert system shell:

1. The rules editor is used to develop the rules that make up the knowledge base. The rules are written in a structured syntax that is then converted into a form usable by the inference engine. In this the rules editor can be compared to a FORTRAN compiler which takes FORTRAN code and converts it into machine language code. One desired characteristic for the editor is that the rules' syntax should be easily understandable in natural language (i.e. English).

2. The user interface is the part of the expert system shell that allows the user to interact with the expert system. The key is that the interface must be user friendly. It must be capable of communicating with both the user and with other programs. The perfect interface would thus be one that uses natural language for input and output, but a menu type interface can also be acceptable. An early lesson in this research was that the user interface must be as flexible as possible.

3. The inference engine is the part of the shell that executes the reasoning algorithms of the expert system. The rules contain the knowledge, and the inference engine applies the knowledge by asking for inputs through the user interface and by making conclusions based on the rules.

4. The knowledge base is analogous to a data base except that the information in the knowledge base is in the form of IF-THEN-ELSE rules. These rules contain the knowledge of the expert system. One or more knowledge bases can be developed using the rules editor and all of the knowledge bases can be interpreted by the inference engine.

Type of Inference Method. The reasoning strategy normally used in any given engineering design problem is forward reasoning. In HVAC design, the facts that are collected at the beginning of the design lead the designer to reason forward towards a completely designed system. The building structure leads to the heating/cooling load calculations. The load calculations lead to equipment capacity, which (along with building type) leads to equipment selection, and so on in a forward manner until the design is complete. In contrast, troubleshooting to find out why a system is not working involves backward reasoning. In troubleshooting, the facts that are collected (such as no air flow or no cooling) lead the repairman to reason backward to discover the cause of the problem.

A clear analogy can be made between forward and backward reasoning and forward and backward chaining (Jackson 1986). The mode of reasoning describes the way in which the program is organized, e.g., forward for design, and backward for troubleshooting. The mode of chaining describes the way the inference engine tests the knowledge base rules. Rules normally take the familiar IF "facts" THEN "outcome" form. The backward chaining system has a built in method for making an initial "outcome" hypothesis. Once the hypothesis is made, the procedure chains backward to see if the "facts" that support the "outcome" match the known input conditions. In forward chaining, the system first looks at the "facts" that match the input conditions and then proceeds to the facts' corresponding outcomes, chaining forward to the final outcome.

Since chaining and reasoning are separate concepts it is possible to implement forward reasoning with both forward and backward chaining. The key to choosing between forward and backward chaining is illustrated in the following comparison. According to Van Horn (1986), backward chaining attempts to minimize the number of questions asked while forward chaining tries to minimize the number of irrelevant possible solutions examined. This implies that the ratio of number of facts to number of outcomes dictates the type of chaining to be used. Many possible outcomes dictates forward chaining while many inputs dictates backward chaining.

In HVAC system design the number of outcomes is or can be made smaller than the number of input facts. This is because the input facts are controlled by the ever changing project environment while the outcome of the design is controlled by the engineer and by the available technology.

The conclusion reached was that the expert system shell used for this project should be a backward chaining shell. To further justify this selection, it should be noted that one of the most commercially successful expert systems in use today is a forward

reasoning, backward chaining system called R1. R1 is used by Digital Equipment Corporation to configure their VAX computer systems. On the surface the tasks of HVAC design and VAX system configuration are very similar.

Type of Certainty Factors. If the user of the expert system does not have a certain input value, he can enter an approximate value. Likewise, if the user does not have even a good approximation for a needed input value the expert system should have a default value that it can use to complete its task. To guard against the possibility of "garbage in gospel out" the expert system should have the capacity of calculating a certainty number to accompany its conclusions. The certainty number should take into consideration the certainty of the user's inputs.

The Dempster-Shafer theory of evidence, of which Bayesian probability and certainty factors (CF) are special cases, is the recommended method for calculating the certainty of a conclusion. This recommendation can be found in both Levine et al. (1986) and in Townsend and Feucht (1986). In summary, the system should:

1. Ask the user to input certain confidence factors along with any data that the user inputs.

2. Assign certainty factors to default data values provided by the system.

3. Calculate the certainty of its conclusions based on the confidence placed on the input data used to reach a conclusion.

Type of User Interface. The ideal user interface is a natural language interface. Current AI research is working towards natural language speech input/output while existing expert systems have written natural language capabilities. The current capabilities of microcomputers, however, make a fully natural language interface impractical.

Although it is recognized that the intended users of this system will have some familiarity with computer use, the user interface should be as close to natural language as possible. The user should not have to read a long manual to learn to use this system. All user inputs should be prompted by the system. In other words, once the program is

initiated the program should lead the user to the final steps. Incorrect entries by the user must not cause the program to "crash." Instead, the system must return to its previous non-error state and ask the user to retry the last input. The expert system must also be capable of telling the user how it reached any given conclusion or why it needs a certain input. In summary, the recommended interface should:

1. Prompt the user for all required user actions. All inputs should be menu driven.

2. Answer specific menu driven questions asked by the user. Examples include telling the user why it needs a certain input or how it made a decision.

3. Return to the previous non-error state and continue from that point forward after the user commits an error.

Separation of Knowledge Base and Inference Engine. The expert system shell should be capable of using knowledge bases that are separate from the inference engine. In other words, the inference engine should be independent of the knowledge base and should be capable of loading a knowledge base from storage into memory. The number of rules in the knowledge base should be limited only by the size of the computer memory and not by the inference engine's capabilities. The reason behind this requirement is that by breaking up the HVAC design knowledge base, computer memory is saved and the resulting expert system is more manageable.

Interface With Other Software and Mathematical Capabilities. Since HVAC design is very dependent on the results of numerical calculations, it would be ideal if the expert system could interface with conventional HVAC design software. In addition to the conventional software available, graphics work stations and HVAC graphics software could be interfaced with the expert system. Several interface options exist. The shell selected for this project should be compatible with as many of the existing options as possible. The possible options are:

1. Direct interface, with the expert system communicating directly with the conventional software or graphics program.

2. Indirect interface, in which the expert system instructs and guides the HVAC designer in the use of the conventional program or graphics program. In other words, the expert system asks the user to run calculations on a given program and return with the results for use by the expert system.

3. Computational programs and/or graphics programs within the expert system program environment. Since AI languages such as LISP have some mathematical capabilities, shells can and do have some mathematics capabilities.

Versatility. This section has discussed the desired attributes of the expert system shell to be selected for this project. These attributes were viewed as the minimum requirements for the development of an HVAC design expert system. Any attributes beyond the minimum were viewed as versatility in the expert system shell. For example, some shells have both forward and backward chaining inference engines, multiple methods of calculating uncertainty and frames capability combined with rule-based capability. Additional attributes beyond the basic were found in some shells, and naturally these shells were considered more useful than shells that met only the minimum requirements.

Shell Selection. A total of 11 AI shell vendors were contacted, of which two did not respond. Table 1 was compiled from the vendors' descriptive data to facilitate comparison of the available expert system shells. As noted earlier, the vendor's descriptive data was not detailed enough to provide a good comparison between the available shells. As most of the top tier shells appeared virtually the same, the shell selection was based strongly on purchase price.

The shell ultimately selected and used in this research is Exsys (Exsys Rel. 3.2.5), a relatively low-cost expert system shell capable of interacting with data base programs and other data producing programs such as psychrometric calculation programs. Exsys is capable of running in either a backward chaining or a forward chaining mode, and the versatility of its user interface has proven very useful in this

Table 1

Comparison of available expert system shells.

SHELL	TYPE OF INFERENCE	TYPE OF PROBABILITY	TYPE OF MENU	KNOWLEDGE BASES	EXTERNAL SOFTWARE INTERFACE
Expert Edge	Backward	Type not specified	Developer defined menu	Not specified	Some data bases number unknown
Insight +2	Backward and forward	Type not specified	Developer defined menu	2000 rules in one base	Up to 3 programs
Exsys	Backward and forward	Three methods type not specified	Developer defined menu	Multiple bases 5000 rules each	Multiple programs with two methods of interface
KES	Backward and forward	Three methods type not specified	Developer defined menu	Multiple bases rules not specified	Multiple programs
PC +	Backward and forward	Type not specified	Developer defined menu	Not specified	Available but not specified
NEXPERT	Backward and forward	Type not specified	Unspecified menu	Multiple bases rules not specified	Multiple programs
ESP Advisor	Backward	None noted	Unspecified menu	Multiple bases rules not specified	Not noted
M.1	Backward and forward	Type not specified	Unspecified menu	Not specified	Not noted
ESIE	Backward	None	Rudimentary menu	Multiple bases rules not specified	None

research. One significant shortcoming of Exsys, however, is that there are no built-in provisions for the user to enter the uncertainty of his input data.

CHAPTER III

STRUCTURE OF HVAC DESIGN EXPERT SYSTEM

General

Figure 3 shows a schematic of the structure of the HVAC design expert system. As can be seen, Exsys is only one part of the complete HVAC design expert system structure. The main feature of this structure is the breakdown of the expert system knowledge base into several smaller knowledge bases.

This feature was based on the assumption that the complete expert system would be a very large program requiring more than the 5000 rule limit Exsys can run on a micro-computer. Splitting the knowledge bases was seen as a method of overcoming this 5000 rule limit.

The validity of this assumption that over 5000 rules will be needed has not yet been proven; however, the fact that the current prototype has over 1000 rules does seem to reinforce the over-5000-rule estimate. Furthermore, even in its current embryonic state the expert system has on occasion exceeded the 640 KB memory limit of the MS-DOS operating system. The combined memory usage of the knowledge bases and the calculation programs they use thus further supports the selection of the multiple knowledge-base structure. In chapter IV the need for this multiple knowledge-base structure will be shown to extend beyond the need for available memory. There it will be shown that this structure is also necessary from a knowledge programming viewpoint. Overall, the structure was conceived as a complete program unit capable of allowing an expert system developed around the Exsys shell to meet the key requirements outlined in

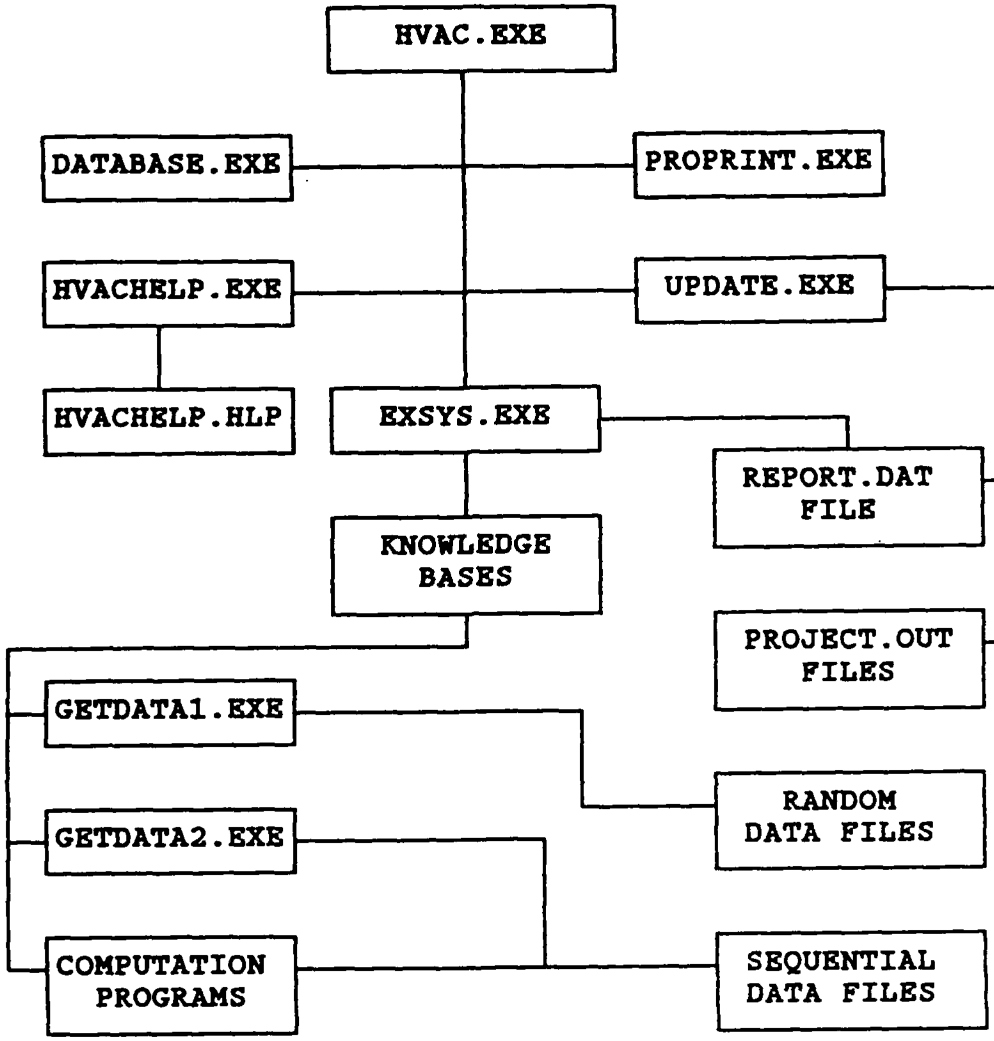

Figure 3

Structure of HVAC design expert system.

the specification of Appendix A. These requirements can be briefly summarized as follows:

a. Separation of design projects with maximum user friendliness.

b. Separation of knowledge bases while allowing all pertinent data to be shared between knowledge bases.

c. Production of a meaningful, presentation-quality project report.

Programming Language

The language used for all the auxiliary programs of the expert system is Quick Basic (Microsoft Quick Basic Rel. 3.0). This compiled language provides several capabilities that were essential for this project.

The first of these capabilities is Quick Basic's ability to call programs using a program statement called "SHELL." This statement, followed by the name of any executable file, has the effect of calling the executable file and halting execution of the calling program. Once execution of the called program is halted, control returns to the calling program at the statement immediately following the "SHELL" statement. This feature makes it possible to control the expert system from one small menu program and thus conserve memory space while maintaining a cohesive, user friendly program unit.

A second useful feature is Quick Basic's ability to concatenate, parse, compare, and otherwise manipulate the values of string variables. The manipulation of the Exsys data input and output at a level not seen by the user is one of the most important features of this expert system structure, and is made possible by the ease with which Quick Basic can work with string variables. It should be noted that LISP's ability to work with non-numerical data is one of its strong points as an AI language.

Finally, the ease of using Quick Basic as compared to C or assembly language supported the selection of Quick Basic over these other languages. Although it is noted that C and assembler are capable of completing the job with faster and smaller code, the learning time for Quick Basic was about 3 hours with prior experience in FORTRAN

programming. The Quick Basic manual (Microsoft Quick Basic Rel 3.0), a text on user interface programming (Simpson 1986) and two magazine articles (Leithauser 1987, 74-75 ; Williams 1987, 54-61) provided most of the subroutines needed to implement the expert system structure using the BASIC programming language.

Programs Used

The programs shown in Figure 3 can be separated into categories as shown in Table 2. Table 2 also includes a list of the files that are used and/or created by each of the listed programs. All programs (except EXSYS) were written as part of this project.

User Interface Programs

HVAC. At the top level of the expert system structure is the HVAC main user interface. This program was written specifically to interact with EXSYS and serves two functions. First, HVAC has an interactive menu that allows the user to work with both EXSYS and auxiliary programs by selecting options from the menu screen. HVAC is the program the user calls at the DOS prompt to begin working with the expert system. By using the menus displayed by HVAC the user can access interactive help screens (HVACHELP program), configure the EXSYS and design project environments (UPDATE program), print a project report (PROPRINT program), and add data to the expert system's random access data files (DATABASE program).

HVAC's second and most important function is its control of the expert system's execution. HVAC prohibits any consultation with EXSYS until the EXSYS and design project environments have been configured by using the UPDATE program. Then, whenever the user does select an EXSYS knowledge base for execution, HVAC calls the UPDATE program both before and after calling EXSYS to insure that the project data is transferred as needed by the selected knowledge base.

HVAC molds the expert system into a cohesive unit by taking care of tedious details in a way that is both user friendly and completely transparent to the user.

Table 2

Program by category and files used by each program.

CATEGORY	PROGRAM	FUNCTION	FILES USED
User interface	HVAC.EXE	Main user interface	FILEPASS.DAT
User interface	HVACHELP.EXE	Help screens	HVACHELP.HLP
Utility	PROPRINT.EXE	Project report	FILEPASS.DAT and PROJECT.OUT
Utility	DATABASE.EXE	Random access file editor	LOCATION.DAT, LOCATION.INX, BUILDING.DAT, and BUILDING.INX
Data access	GETDATA1.EXE	Access random files	INPUT.DAT, plus all files listed for DATABASE.EXE
Data access	GETDATA2.EXE	Access sequential files	Sequential files and INPUT.DAT
Data management	UPDATE.EXE	Update project files	EXSYS.CFG, REPORT.CFG, FILEPASS.DAT, REPORT.DAT, PROJECT.OUT and HOLD.DAT
Expert system shell	EXSYS.EXE	Inference engine	EXSYS.CFG, REPORT.CFG, REPORT.DAT, OUTPUT.DAT, INPUT.DAT and knowledge bases i.e. MAINKB1.RUL/.TXT and SYSTKB1.RUL/.TXT
Computation	PSYCH1.EXE	Psychrometrics	INPUT.DAT
Computation	PSYCH2.EXE	Psychrometrics	INPUT.DAT
Computation	PSYCH3.EXE	Psychrometrics	INPUT.DAT
Computation	STD90-80.EXE	ASHRAE Standard 90-80 curve fit	INPUT.DAT

HVACHELP. As the name implies, HVACHELP is an interactive help screen program. This function is kept separate from the HVAC program to conserve computer memory and also to allow it to be used from other utility programs of the expert system. HVACHELP allows the user to view the help messages contained in the HVACHELP.HLP file. This file is a random access file containing help information for EXSYS and for the other programs of the expert system. HVACHELP allows the user to page back and forth through the HVACHELP.HLP file. It also allows the user to go directly to the sections of HVACHELP.HLP that he needs to see without requiring him to page through information he does not currently need. Thus, HVACHELP provides interactive help to the user consistent with the desired user friendly characteristics of the expert system.

Utility Programs

PROPRINT. As noted above, the HVAC program serves as a menu for working with EXSYS and some of the utility programs of the expert system. The first of these utility programs is PROPRINT.

Although EXSYS provides a user-programmable report generation feature, the reports possible with EXSYS alone are not very clear. PROPRINT is essentially a text processor capable of taking the EXSYS output and producing a meaningful report. When PROPRINT is called it first reads the FILEPASS.DAT file, which contains the name of the currently active project file. It next provides the user with the option of selecting a different project for printing, then proceeds to read and print the contents of the correct project data file.

The functions and subroutines that make up the PROPRINT program have all been debugged and tested with good results, and the program has been sufficiently completed to provide a sample report from the data and results of the first four knowledge bases. This small amount of output is sufficient to demonstrate the capabilities of the

program and to show its advantages over the EXSYS report generator. The program, however, has not been updated to read and print the outputs currently developed by all of the knowledge bases.

DATABASE. The other auxiliary program currently accessible through HVAC is DATABASE. This program is a simple data base editor for random access files. With DATABASE the user can edit and sort random access data files for use with the GETDATA1 program (GETDATA1 and a similar program called GETDATA2 will be explained later).

DATABASE (and GETDATA1) use two files for each database: the ".DAT" files (LOCATION.DAT and BUILDING.DAT) and the ".INX" files (LOCATION.INX and BUILDING.INX). The ".DAT" files are random access files that contain the actual data, and the ".INX" files are sequential access files used to index the data in the corresponding ".DAT" files.

Data Access Programs

HVAC design is a process that requires much data. Because the expert system should be capable of doing everything the human expert is capable of doing, it is necessary to give the expert system the capability of finding needed data without having to ask the user for data that the human expert could find for himself. The DATABASE program explained above is one of the programs developed to give the expert system a data access capability, but the actual data access programs are GETDATA1 and GETDATA2.

When the user selects a knowledge base from the HVAC menu, program control is turned over to EXSYS. EXSYS loads the selected knowledge base and the inference engine begins to scan the rules in the knowledge base, asking for user inputs or accessing the data bases as needed. When EXSYS requires data contained in the expert system's data bases, GETDATA1 or GETDATA2 is called, temporarily halting the EXSYS

execution and turning control to GETDATA1 or GETDATA2. Once the data are found, GETDATA1 or GETDATA2 sends the data to EXSYS in the INPUT.DAT file, ends execution, and returns control to EXSYS.

GETDATA1. EXSYS has three data types; *variables, qualifiers* and *choices*. *Variables* and *qualifiers* will now be explained. Variables are very much like the variables normally found in FORTRAN and BASIC. In EXSYS, variables are of two types; string and numeric. As in FORTRAN and BASIC the user can enter any value for variables when asked to do so. The exceptions to this are that numeric variables will not accept strings nor will they accept out of range numeric inputs. The programer predetermines the ranges for EXSYS numeric variables.

Unlike the variable, the qualifier is a data type that only accepts pre-determined inputs selected from a menu. The text of qualifiers end in verbs and are followed by one or more values when the values are assigned. An example of a qualifier and some of its values is as follows:

QUALIFIER	=	the location is
VALUES	=	1. GAINESVILLE_FL
		2. CHICAGO_IL
		3. DAYTON_OH

The qualifier with a value assigned would read: "the location is GAINESVILLE_FL." The advantage of the qualifier is that it allows the programmer to limit the user's inputs to those inputs that the expert system can understand. In the above example, the user is prevented from entering a location unknown to the expert system. Unfortunately, EXSYS has a limit of 30 input values per qualifier.

In programming the expert system it became evident that there are more than 30 building types (if buildings are classied by function i.e., residence, offices, churches, etc.). Likewise, it would be advantageous to allow the user to choose project locations from more than 30 geographic locations. GETDATA1 was developed to restrict the inputs to string variables as the inputs to qualifiers are restricted by EXSYS, thereby

expanding the 30 building type and 30 location limit. When the expert system needs to know the building type, GETDATA1 is called to access the BUILDING.DAT file. GETDATA1 is interactive and displays a list of 47 building types from which the user may choose. When the user makes his selection GETDATA1 returns the value to EXSYS. Similarly, GETDATA1 returns the project location to EXSYS by accessing the LOCATION.DAT file.

With the DATABASE program the user can add values to the LOCATION.DAT and BUILDING.DAT files. However, adding new building types to the BUILDING.DAT file without updating the knowledge bases to accept the new building types will result in no output from the knowledge bases. A possible solution to this problem will be discussed in Chapter VI.

GETDATA2. GETDATA1 is used to allow the user to send specific data to EXSYS and to work around the 30 qualifier value limit. GETDATA2, on the other hand, is used to send data to EXSYS without user interaction. When EXSYS needs information (such as weather data or the per ton cost for a particular HVAC system) it calls GETDATA2 to search sequential data files until it finds the needed data. The data are then returned to EXSYS without requiring user inputs. This method of passing data to EXSYS saves space in the knowledge bases and makes the expert system less dependent on user inputs. A text editor capable of producing unformatted ASCII files is used to build and maintain the data files for GETDATA2.

Data Management Program

UPDATE. UPDATE is the only program under this category and is the central nervous system of this expert system structure. The operation of UPDATE is controlled by the HVAC program; that is to say that HVAC calls UPDATE at the appropriate times and with the appropriate arguments. Thus, the use of UPDATE is transparent to the expert system user. Although the functions in UPDATE could have been built into the

HVAC program, keeping UPDATE separate from HVAC saves memory as UPDATE does not have to be memory resident while EXSYS is running. UPDATE operates in three separate modes and selects its mode of operation based on the command line arguments used to call it. The command line is: "UPDATE [kb] [mode]."

In the command line, "[mode]" is a one letter code that tells UPDATE what mode to run in. Code letters are "P," "S," and "F." When the mode is "P" the "[kb]" in the command line is the word "SPACE." When the mode is "S" or "F" the "[kb]" in the command line is one of the following two character codes: M1, M2, M3, S1, S2, S3. These two character codes identify to UPDATE the knowledge base that is being used. Three sample calls to UPDATE are:

UPDATE	SPACE	P
UPDATE	M1	S
UPDATE	S1	F

UPDATE's first function is its configuration of the active project environment. This is the "P" mode and it takes place when the user selects "1. SELECT/START PROJECT" from the main menu of HVAC. On this selection HVAC's call to UPDATE is: SHELL "UPDATE SPACE P." This function involves three separate steps.

The first step is selection or initiation of the active project file. Through UPDATE, the user selects or builds a project file for each of his design projects. These files have the extension ".OUT" and are represented in Table 2 and Figure 3 by the "PROJECT.OUT" file. To select an existing file the user enters the name of the file. If the file exists on the current directory UPDATE places the name of the selected file in the "FILEPASS.DAT" file. The FILEPASS.DAT file makes the name of the selected project file available to the programs of the expert system.

To create a project file the user enters a new file name. UPDATE, not finding the file name on disk, creates a file with the name the user provided, and puts into it the 13 labels shown in Figure 4. Figure 4 is a representation of the empty project file. The M1, M2, M3, S1, S2, and S3 labels mark the beginning of six data sections in the project file.

The "@" symbol marks the end of each section and the LL label marks the end of the file. Each data section will later be used to separately store data from the knowledge bases. After creating the new file, UPDATE puts the name of the file in the FILEPASS.DAT file as it did when an existing project file was selected.

The second step in the UPDATE program's configuration mode is EXSYS configuration. After selecting or creating a new project file, UPDATE opens a file called EXSYS.CFG. This file configures EXSYS to behave in the following way. First, it tells EXSYS to use a file called REPORT.DAT for input of general project data. Second, it tells EXSYS to format its output according to the instructions in a file called REPORT.CFG. Third, it tells EXSYS to use a file called OUTPUT.DAT to send data to external calculation programs. Fourth, it tells EXSYS to receive the results from these external programs in a file called INPUT.DAT. Last, EXSYS.CFG tells EXSYS to conserve computer memory by not loading the display text associated with the knowledge-base rules into memory until the text is actually needed.

The third and final step in UPDATE's configuration mode is EXSYS report configuration. Here UPDATE opens the REPORT.CFG file that was noted above. This file instructs EXSYS to send its reports to the REPORT.DAT file in a specific format. This format, which is easily read and manipulated using the PROPRINT and UPDATE programs, is explained later in this section. Once this final step is complete, UPDATE ends execution and returns control to the HVAC program, making it possible for the user to select the expert system's knowledge bases.

The second mode of operation for UPDATE is the "S" mode and it takes place when the user selects a knowledge base from the system design menu of HVAC. When the user selects a knowledge base, HVAC first calls UPDATE with the following command line: "UPDATE [kb] S." Recall that [kb] is one of the codes M1, M2, S1 etc. and that each code refers to one of the knowledge bases in the expert system. Thus if the

```
M1
@
M2
@
M3
@
S1
@
S2
@
S3
@
LL
```

Figure 4

Format of empty PROJECT.OUT file.

user is selecting the preliminary design or M1 knowledge base the call to UPDATE is: "UPDATE M1 S."

The "S" in the command line tells UPDATE to operate in the "start" mode. In this mode UPDATE begins by reading the name of the active project file from the FILEPASS.DAT file. Then the "M1" in the command line directs UPDATE to read all data located in the "M1" section of the active project file (see Figure 4). If the active project file is new and no data is found in the M1 section UPDATE ends execution. If UPDATE finds data in the M1 section it reads it and puts it in a file called REPORT.DAT. Recall that REPORT.DAT is the file EXSYS uses for input and output of general project information. Once UPDATE has transferred all pertinent data from the active project file to the REPORT.DAT file it ends execution, vacates computer memory and returns control back to HVAC.

Let us again assume that the user has selected the preliminary design knowledge base from the menu in HVAC. In response to the user's selection HVAC called the UPDATE program as was explained above, and UPDATE returned control to HVAC. Without further user input HVAC now calls EXSYS to operate on the preliminary design knowledge base. EXSYS first reads the data in the REPORT.DAT file and then executes the selected knowledge base, asking for user inputs and using the data gathering and calculation programs as necessary. When EXSYS ends execution it sends all of its output to the REPORT.DAT file.

Instead of replacing the data in the REPORT.DAT file with the new data, EXSYS adds the new data to the end of the file leaving the old data intact. Figure 5 illustrates a REPORT.DAT file before EXSYS execution. Figure 6 illustrates the same REPORT.DAT file as Figure 5, but Figure 6 is an illustration of the file after EXSYS execution. Note that EXSYS has added data to the end of the file and has left the original data undisturbed. After writing its output to the REPORT.DAT file EXSYS ends execution, vacates computer memory, and returns control to the HVAC program.

As soon as EXSYS ends execution, the HVAC program calls the UPDATE program to store the results of the EXSYS consultation in the active project file. This is the third and final mode of operation for the UPDATE program. If we still assume that the user has been working with the preliminary design knowledge base the call to UPDATE is: "UPDATE M1 F."

As before, the "M1" code tells UPDATE that it is working with the M1 section of the active project file. The "F" code tells UPDATE to operate in the "finish" mode. In this mode UPDATE begins by reading the FILEPASS.DAT file to determine the name of the active project file. Then UPDATE proceeds to update the data in the project file using three distinct procedures. It reads the PROJECT.DAT file, performs a data conversion to make all data types useful to EXSYS, and updates the data in the active project file. Explanations of these three procedures follow.

First UPDATE reads the last set of data in the REPORT.DAT file. This is the latest data that EXSYS has written to the file. In Figure 6 this last set of data begins on line number eight which reads "C1 VAV: Probability=93/100." UPDATE discards all earlier data and keeps only this last set.

In the next step UPDATE performs a data-type conversion on some of the data it has just read so that the results from the currently selected knowledge base can be used in the other knowledge bases of the expert system. To understand this procedure we must first understand all the EXSYS data types.

Recall that EXSYS has three data types. The *variable* and *qualifier* data types were both explained in the section on the GETDATA1 program. Figure 5 illustrates a REPORT.DAT file that contains variables and qualifiers. The first three lines are qualifier values. The Q3, Q5, and Q6 identify the qualifiers while the numbers that follow are the value (or values) of each qualifier. The remaining data shown in Figure 5 are variables. The V1, V2, etc. identify the variables while the numbers or alphanumeric strings that

```
Q3 4,5
Q5 2
Q6 2
V1 OFFICE_LOW_RISE
V2 DALLAS_TX
V5 6.000000
V7 13050000.000000
```

Figure 5

Sample REPORT.DAT file before call to EXSYS.

```
Q3 4,5
Q5 2
Q6 2
V1 OFFICE_LOW_RISE
V2 DALLAS_TX
V5 6.000000
V7 13050000.000000
C1 VAV: Probability=93/100
C4 Multizone: Probability=55/100
C6 Fan-Coil Units: Probability=67/100
Q1 3
Q2 3
Q3 4,5
Q5 2
Q6 2
V1 OFFICE_LOW_RISE
V2 DALLAS_TX
V5 6.000000
V7 13050000.000000
V9 12.000000
```

Figure 6

Sample REPORT.DAT file after call to EXSYS.

follow are the values of each variable. In Figure 6 we also see variables and qualifiers but we also see three lines that do not begin with either a "Q" or a "V." These are:

C1 VAV: Probability=93/100
C4 Multizone: Probability=55/100
C6 Fan-Coil Units: Probability=67/100

In EXSYS terminology these are choices. Choices are the *results* of a consultation with an EXSYS knowledge base. The knowledge base looks at the known facts (variables and qualifiers), and based on the facts and on the knowledge programmed in the rules assigns probability values to each of the choices programmed in the knowledge base. Thus when EXSYS writes the results of a consultation to the REPORT.DAT file it writes a probability value after the text or value of each choice. These two elements are seen in Figure 7 where the text and probability value of the three choices from Figure 6 are separated and labeled.

Within this prototype expert system it was necessary to use the results (choices) of one knowledge base as data in a second knowledge base. For example, one knowledge base selects a HVAC system for a given project, then the next knowledge base selects a control system based on the HVAC system that has been selected.

Unfortunately EXSYS does not provide a built-in method for passing choices from one knowledge base to another. If we leave choices and their probabilities in the format shown in Figure 6 a read error will occur when EXSYS tries to read the REPORT.DAT file. Therefore, to make choices useful as data, the UPDATE program converts each choice and probability pair into two variables. The results of such a conversion using the choices from Figure 6 would look as follows:

V10 VAV
V11 93
V12 Fan-Coil Units
V13 67
V14 Multizone
V15 55

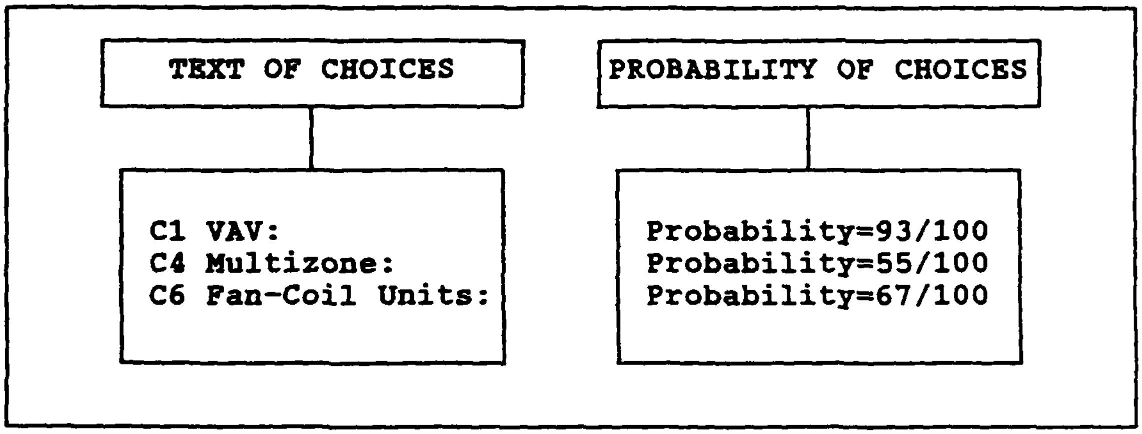

Figure 7

Values and probabilities of choices shown in Figure 6

Notice that variables V10, V12, and V14 contain the text of the choices while variables V11, V13, and V15 contain the probability values. Also notice that in the conversion process the probabilities are converted into an integer value. What was "Probability=93/100" is reduced to "93" in the final product. Finally, note that the labels C1, C4, and C6 that preceded the text of the choices have been discarded since they are only needed to implement the data conversion process.

In the third and final step UPDATE writes all of the latest data (including the converted choices) to the active project file. UPDATE does this by first putting all the data in the section for the selected knowledge base. Next, UPDATE cross references the data from the selected knowledge base to the other knowledge bases and updates the data sections for all the other knowledge bases. In this way all data gathered or determined by a knowledge base is available to all other knowledge bases in the expert system.

This process of updating data in the sections of the active project file can be implemented in both up and down channel directions. In down channel updating the data from a knowledge base is passed to all knowledge bases that will be consulted later in the design process. In up channel updating the data from a knowledge base is passed to all knowledge bases that would have been consulted earlier in the design process. Down channel updating is clearly the most desirable option and is the method used in the current version of the UPDATE program. Although the option of conducting both up and down channel updating exists, it has been only marginally successful and little more advantageous than down channel updating alone.

Computation Programs

The programs in this group are engineering computation programs that have been written specifically for this expert system. An alternative approach could have been to write intermediate programs to interface commercial HVAC design programs with the

EXSYS expert system shell. This second approach was not used because available programs are too general and too large for use in this expert system.

The programs described in this section are called by EXSYS when EXSYS needs values that can be calculated by the programs. This is done using built-in EXSYS functions. In general, EXSYS passes data to these programs either in the command line or in a file called OUTPUT.DAT. Once a program is called, control is turned over to the program and EXSYS waits while the program completes its calculations. In some cases the data passed by EXSYS to a program are not sufficient for the program to complete its calculations. If so, it is then possible for the program to ask the user for inputs; or more commonly, the program can read data from one or more sequential data files using the same algorithms used in the GETDATA2 program. Once the program completes its calculations it sends its results to EXSYS in a file called INPUT.DAT, ends execution, and returns control to EXSYS. The format for data returned in the INPUT.DAT file is shown in Figure 8. In Figure 8, V15, V16, V17, and V18 are the variable labels used to tell EXSYS what variable number each datum is for. The 3.9, 3.5, etc. are the data being passed to EXSYS.

The following sections contain brief descriptions of the computation programs that have been developed for the expert system thus far:

PSYCH1. This psychrometric program considers the effects of elevation when it calculates the percent relative humidity for air at the inside design dry-bulb temperature and the 2.5% outside design humidity ratio. The outside design humidity ratio is as determined by the 2.5% outside design dry-bulb temperature and the 2.5% mean coincident wet-bulb temperature.

PSYCH2. This psychrometric program calculates percent relative humidity for air with conditions of 2.5% outside design dry-bulb temperature and 2.5% mean coincident wet-bulb temperature. The effects of elevation are considered.

```
V15 3.9
V16 3.45
V17 0.2333
V18 0.050000
```

Figure 8

Sample INPUT.DAT file.

PSYCH3. Similar to PSYCH1, this program calculates percent relative humidity for air at 70 ^{o}F (indoor temperature) and the average outdoor winter humidity ratio. The average outdoor winter humidity ratio is as determined by the 97.5% outside design dry-bulb temperature and the average relative humidity for January.

STD90-80. This curve fit program returns values from functions that curve fit the 12 graphs found in ASHRAE standard 90-1980 (ASHRAE 1980). This standard, entitled Energy Conservation in New Building Design, provides design guidance for the selection of energy efficient levels of lighting and insulation in buildings. The inputs to this program vary depending upon the function used, but include the building type code from ASHRAE standard 90-1980, heating degree days, building height in stories, and degree North latitude. The building type code is determined by the expert system and is essentially used to classify buildings as residential or commercial.

Expert System Shell

EXSYS. EXSYS is the only program in this expert system package that was not written as part of this project. As noted, EXSYS is a commercially available expert system shell. The software package that comes with EXSYS includes the EXSYS inference engine and user interface and the EXSYS knowledge-base editor. With the editor the expert system developer builds knowledge bases made up of IF-THEN-ELSE rules. Then with the inference engine the user consults the knowledge bases. The programs written for this expert system allow EXSYS to access HVAC data both from data bases and from engineering computations, and also allow for the passing of data between the different knowledge bases.

EXSYS interacts with the user by asking for data inputs and by displaying logic paths and known data at the user's request. EXSYS displays logic paths by showing the user the rules in the knowledge base. One way it does this is by displaying the applicable rules in reverse order.

In this expert system EXSYS is being used in a backwards chaining mode. This means that EXSYS begins with a result ("choice" in EXSYS terminology) and attempts to validate that result by looking at the facts that make that choice true. EXSYS does this for all possible results in a knowledge base. When EXSYS displays rules in reverse order it begins by displaying the rule it is currently looking at and takes the user from the facts to the results. EXSYS can also display the logic from the results to the facts that support the results. A few sample rules can help to clarify these points:

Rule 1.
IF (1) [VARIABLE 1]=A
THEN (1) [VARIABLE 2] IS GIVEN THE VALUE "B"

Rule 2.
IF (1) [VARIABLE 2]=B
THEN (1) [VARIABLE 3] IS GIVEN THE VALUE "C"

Rule 3.
IF (1) [VARIABLE 3]=C
THEN (1) USE MULTIZONE: Probability=90/100

In these rules EXSYS wants to prove rule 3 (USE MULTIZONE: Probability=90/100) so it backward chains to rule 2 because the information needed to prove rule 3 can be found by proving rule 2. Next it backward chains to rule 1 in order to prove rule 2. If it cannot find another rule that can help prove rule 2, EXSYS will ask the user for an input. In this case EXSYS asks the user for the value of [VARIABLE 1]. The user can enter a value or he can ask EXSYS why it wants the requested value. In answer EXSYS displays rules 1, 2 and 3 in that order telling the user why it needs the requested information.

Using the same set of sample rules, assume the user enters "A" to EXSYS' request for a value for [VARIABLE 1]. The result that EXSYS will reach in such a case is "USE MULTIZONE: Probability=90/100." If at this point the user asks EXSYS to display how it arrived at this conclusion, EXSYS displays rules 3, 2, and 1 in that order. If the user asks EXSYS how [VARIABLE 1] got the value "A" EXSYS responds "You told me," reminding the user of his data entry.

Using Other Expert System Shells

As has been shown, the expert system shell, EXSYS in this case, is only one part of the expert system package. The structure and programs described in this chapter were designed both to adapt EXSYS to HVAC design and to eliminate some of the weaknesses in EXSYS.

Because of the complex nature of HVAC design as compared to most of the knowledge domains to which expert systems have been applied, it is doubtful that any generic expert system shell can yield a stand-alone HVAC design expert system without the aid of auxiliary programs. Building an HVAC design expert system using more advanced expert system shells will certainly reduce the need for auxiliary programs, while using simpler shells will increase the need for such programs. In any event, alternative structures and programs would have to be developed when building similar expert systems around other expert system shells.

CHAPTER IV

KNOWLEDGE PROGRAMMING

Overview

Both the experience of programming the knowledge bases of this expert system and the published experiences of other programmers (Brothers, 1988) strongly indicate that the process of programming a knowledge base is an interactive and iterative process. To be effective, the programmer must continually switch between the processes of knowledge acquisition and knowledge programming. Furthermore the overall process is evolutionary and in most cases the finished knowledge bases are very different from the knowledge bases that were initially conceived.

As new knowledge was added during the course of this project, new and better rule structures were discovered. This, as can be expected, led to several changes not only in the way the rules were arranged within the knowledge bases but also in the structure and arrangement of the knowledge bases. Some knowledge bases were discarded because the required changes made it easier to start anew, while others were only changed.

To a novice knowledge programmer who has only limited programming experience using conventional languages such as FORTRAN, the iterative nature of knowledge programming can be very disturbing. First of all the number of changes seem to hamper progress and the programming project appears to be standing still. Second, in most fields of human endeavor frequent changes or deviations from a chosen path tend to indicate poor planning or lack of forethought. A recent paper by P. W. Brothers (1988), however, explains that in knowledge base programming changes in programming

direction are the norm. This, Brothers indicates is especially true during the early stages of program development.

The iterative and evolutionary development process described by Brothers is similar to the process that took place during the course of this project where two distinct programming phases occurred. These phases can be classified as "familiarization with EXSYS," and "evolution of the knowledge bases." The former will be discussed here while the latter is the topic of the following chapter.

Familiarization With EXSYS

The initial knowledge programming work that was done for this project can be classified as familiarization with the EXSYS expert system shell. As can be expected, studying the EXSYS manual marked the beginning of this phase. Familiarization with the manual was followed by the programming of about 50 rules in each of two knowledge bases. This phase also included the development and testing of the auxiliary programs of the expert system, an accomplishment which proved the feasibility of the expert system structure. In addition to validating the expert system structure, this early phase of programming provided a detailed understanding of the EXSYS logic (inference) algorithm.

EXSYS Data Types

The first step in learning to program knowledge with EXSYS is to learn all the different options available when using the three EXSYS data types.

The first data type is the variable. In the EXSYS syntax the variable is denoted by a suitable variable name placed inside of square brackets; i.e. [D-B TEMPERATURE]. EXSYS variables can be numeric or string variables, and their display at the end of an EXSYS run is an important programmer-selected option that affects the way in which EXSYS rules are tested by the inference engine. In addition to the variable name, EXSYS variables can have a variable description associated with each variable. These

descriptions are used by the EXSYS user interface when EXSYS refers to its variables. For example, assume that the description for the variable [D-B TEMPERATURE] is "*the interior design dry-bulb temperature in degrees F.*" When EXSYS requires an input for the [D-B TEMPERATURE] variable the user interface will display:

Please input the interior design dry-bulb temperature in degrees F

Assuming that the above variable is to be displayed at the end of an EXSYS run, and assuming that it has been given the value of 70, the display for the variable at the end of an EXSYS run would be:

The interior design dry-bulb temperature in degrees F = 70

It should be noted that the variable's description is a device that allows the user interface to communicate with the user in a manner more natural than the usual alternative, which is for the program to display a message such as: "Please enter [D-B TEMPERATURE]."

The second EXSYS data type is the qualifier. The qualifier is composed of two parts --the text of the qualifier and its associated values. For example, the text of a qualifier and its associated values would be:

Text: The occupants' level of activity is

Values:
1. sedentary
2. office work
3. walking
4. light recreational exercise
5. heavy work or exercise

The EXSYS manual claims that the text of a qualifier must end in a verb. Experience, however, has shown that this is not true, and that the only real requirement is that the text of the qualifier and its associated values agree grammatically and form a meaningful, easy to understand statement. This requirement is illustrative of the advantage of using qualifiers; namely that condition statements written with qualifiers are much clearer and easier to understand than condition statements written with variables.

When EXSYS requires an input to a qualifier, it displays the text of the qualifier followed by the associated values. Still using the above sample qualifier as an example, the EXSYS display would look as follows when EXSYS requests an input to a qualifier:

The occupants' level of activity is
1 sedentary
2 office work
3 walking
4 light recreational exercise
5 heavy work or exercise

From the above display the user would make a selection by entering the number of the choice followed by the return key. In the event that two or more responses are appropriate the user can enter multiple responses by entering a number followed by a comma followed by the next number and so on. This illustrates a second significant advantage to using qualifiers instead of variables as variables can only have one value assigned to them at a time.

To improve the syntax of qualifiers EXSYS provides two additions to the basic format of the qualifier. First it allows the programmer to insert value text in the middle or beginning of qualifiers. For example, the following is the same qualifier written in two different forms:

The main purpose for HVAC in this building is
1. comfort
2. health care
3. archival storage
4. to support a process

In this building ____ is the main purpose for HVAC
1. comfort
2. health care
3. archival storage
4. process support

The underline in the middle of the second example holds a place for the value text and when a value is chosen, "comfort" for example, the underline is replaced by the value text resulting in: "In this building comfort is the main purpose for HVAC." The added latitude afforded when writing qualifiers is the clear benefit of this addition.

The second modification that EXSYS provides for is the insertion of a variable in the text of the qualifier. In the above example it is possible to add the variable "[BUILDING TYPE]" to improve the user interface capabilities:

In [[BUILDING TYPE]] ____ is the main purpose for HVAC
1. comfort
2. health care
3. archival storage
4. process support

Assuming the variable [BUILDING TYPE] has the value "hotels" and the qualifier the value "comfort," the user interface will display "In hotels comfort is the main purpose for HVAC." This option is useful for enhancing the user interface but does not help in rules where the value of the embedded variable is not shown. In a rule the above qualifier may appear as follows:

IF: (1) [BUILDING TYPE] = "hotels"
THEN: (1) In [[BUILDING TYPE]] comfort is the main purpose for HVAC

As noted in chapter III the major limitation to EXSYS qualifiers is that they have a limit of 30 possible answer values per qualifier. Recall that the GETDATA1 program was written to overcome this limitation.

The third EXSYS data type is the choice. Choices are the results of an EXSYS run. In other words, they are solutions to the problems that are being solved by the expert system. An example of a choice is:

Use 2.5% design dry-bulb temperature.

The above example is a valid EXSYS choice and has the correct syntax; however, in this project an additional requirement has been placed on the syntax for choices. To make it possible for the UPDATE program to manipulate choices it was necessary to add an identifier to each choice. The identifier used is the upper case letter "C" immediately followed by a number. Thus in this project the above sample choice must be written as:

C1 Use 2.5% design dry-bulb temperature.

In each knowledge base the choices are numbered consecutively and each knowledge base begins with choice number one. A different numbering scheme would be acceptable as long as choice numbers are not duplicated in any one knowledge base. Six choices labeled "C1" are acceptable as long as each is in a different knowledge base because the UPDATE program has the capability to differentiate between like numbered choices from different knowledge bases.

EXSYS Rules

The rules that make up the EXSYS knowledge bases have two parts, the head or "IF" part of the rule and the tail or "THEN/ELSE" part. In general the syntax of these rules is the familiar antecedent and consequent type logic statement. Nuances within this general syntax, however, warrant an explanation of the EXSYS rules.

Head of Rules. The head of an EXSYS rule contains the antecedent conditions for the rule. Here EXSYS allows logical comparisons with variables. For both numeric and string variables the allowable logical operators are =, <, >, <=, >=, and <>. Respectively these operators are equal, less than, greater than, less than or equal, greater than or equal and not equal. In string variables the comparison is based on alphabetical order, and in this project only the equal and not equal operators have been used for string variables. The following is an example of a variable at the head of a rule:

IF: (1) [BUILDING TYPE] = "SUPERMARKET"

In the above example note that a number one preceded the antecedent statement. Multiple antecedent statements are allowable within each rule and EXSYS enumerates each of the conditions by preceding them with a number.

The head of EXSYS rules can also contain qualifiers and combinations of both qualifiers and variables. A rule containing a qualifier as the antecedent statement is:

IF: (1) the occupants' level of activity is sedentary

Recalling that it is possible for the user to enter more than one value for qualifiers, note that it is possible to have multiple qualifier values in the head of a rule. This observation leads to some variations of the "IF" statement. Entering two or more values for one qualifier results in introducing an "OR" operator into the rule. An example of this is as follows:

IF: (1) the occupants' level of activity is sedentary, or office work or walking

It is also possible to enter the same qualifier in more than one condition for the same rule, resulting in the introduction of an "AND" operator into the rule. An example of this type of rule follows. Note how the multiple conditions are numbered:

IF: (1) the occupant's level of activity is sedentary
and (2) the occupant's level of activity is office work

A serious flaw with EXSYS is that it is not possible to introduce an "OR" operator for a condition that contains a variable. The only method found to circumvent this flaw was to write multiple rules either with identical tails or with chaining between the rules. For example, to write a rule that makes the comparison "IF [BUILDING TYPE] = HOTEL or MOTEL," two rules could be written making sure that the tail or "THEN" part of the rules are the same. If for the two following rules the "THEN" statements are the same, the effect of the two rules is the same as having an "OR" operator, i.e. "MOTEL or HOTEL." The heads of the rules would then be:

Rule 1. IF: (1) [BUILDING TYPE] = "MOTEL"

Rule 2. IF: (1) [BUILDING TYPE] = "HOTEL"

It is also possible to circumvent the problem by grouping data. Using the same example, multiple rules could be used to classify hotels, motels, and dormitories as domiciliary type buildings using a qualifier. Then when it is necessary to have "HOTEL or MOTEL" in the logic, one can write one rule with the following head which includes "HOTEL or MOTEL" and excludes "DORMITORY":

IF: (1) Classification is domiciliary
and (2) [BUILDING TYPE] <> "DORMITORY"

Unlike the "OR" operator, introducing an "AND" operator for variables presents no significant problems. The process is the same as introducing an "AND" operator using qualifiers; that is the variable is used in more than one antecedent statement in the same rule. For example, to program 70 < [D-B TEMP] <= 80, one would enter two conditions in the same rule:

IF: (1) [D-B TEMP] <= 80
and (2) [D-B TEMP] > 70

<u>Tail of Rules.</u> The tail of EXSYS rules may contain "THEN" and "ELSE" clauses in any combination (i.e. "THEN" only, "ELSE" only and both "THEN" and "ELSE"). Within both types of clauses several different options are allowed. The variable, the qualifier, and the choice are all used differently in the consequent part of the rules.

First, the "THEN" part of the EXSYS rules can be used to assign values to string variables and numeric variables. With numeric variables the assigned values can be a constant or an algebraic equation of limited scope. The algebraic equations are limited to 20 sets of parentheses and 100 total characters per equation. The former limit does not present a problem; the latter, however, is very restrictive because for variable names to be meaningful they often need to be long. Thus three or four variables with adequately descriptive names (i.e. [2.5% D-B TEMP]) may exceed the 100 character per equation limit. For mathematical operations EXSYS provides trigonometric, exponential, and logarithmic functions as well as the four arithmetic operations.

The following sample rule illustrates the use of variables in the tail of EXSYS rules. Note that just as it is possible to have more than one antecedent condition in the head of rules, it is possible to have more than one consequent condition in the tail of the rules. Also note that EXSYS uses the operator "IS GIVEN THE VALUE" instead of the usual equal sign to assign values to variables:

THEN: (1) [BUILDING TYPE] IS GIVEN THE VALUE "MOTEL"
and (2) [INT D-B TEMP] IS GIVEN THE VALUE 70
and (3) [TOTAL AREA] IS GIVEN THE VALUE [FLOOR AREA]* [NUMBER OF FLOORS]

As with variables, the tail of EXSYS rules can be used to assign values to qualifiers. The syntax is handled automatically by the EXSYS rules editor. Essentially the text of the qualifiers and the value(s) to be assigned are entered from a menu of available qualifiers. The following is an example. Note that to assign multiple values it is not necessary to use multiple consequent conditions:

THEN: (1) the occupant's level of activity is sedentary
and office work

As has been noted, the third EXSYS data type, the choice, can only appear at the tail of EXSYS rules. The syntax for choices is always the same --the text (or value) of the choice appears followed by a probability value:

THEN: (1) C1 Use VAV system - Probability=45/100

In the above syntax, the text "Use VAV system" is the useful output from the knowledge base. The probability, in this example 45/100 or 45%, is the certainty value that the sample rule applies to the result. In other words the rules, when the antecedent condition(s) are true, assign to each choice a predetermined probability value. That probability value should indicate the confidence that the domain expert has in the value of the choices being true, based on the given the antecedent conditions of the rule. Thus if an antecedent condition is necessary and sufficient the probability assigned should be 100%, and in all other cases where the antecedent is not necessary or sufficient the probability should be less than 100%.

All that has thus far been described about the syntax of the "THEN" statements is applicable to the "ELSE" statements. In this project the "ELSE" statement has only been used in a few rules, and by all indications it appears that with EXSYS significant use of the "ELSE" statement can only take place in knowledge bases with very narrow knowledge domains.

The "ELSE" statements of a rule execute anytime that the head of the rules is false. Because of this the cautions against the use of the "ELSE" statement are analogous to the

cautions against the use of words like "never" and "always." The problem is that the "ELSE" statement must be applicable anytime the head of the rule is false. For example, rules with heads such as: "IF [BUILDING TYPE] = "RESIDENCE" cannot, in general, have an "ELSE" statement. The "ELSE" statement would be executed anytime that the building is a not a residence, so it would execute both when the building is a supermarket and when it is an office building, and clearly a supermarket and an office building are very different facilities. Thus it would be hard to find an "ELSE" statement applicable to both supermarkets and office buildings, not to mention the 40 or so other building types programmed into this expert system. Thus "ELSE" statements were rarely used in this project.

Methods For Combining Probabilities

In EXSYS there are five different methods to account for the probabilities assigned by the THEN/ELSE statements of the rules. The purpose of each of these methods is to account for and combine all of the probabilities assigned to every choice in the knowledge base by applicable rules.

The particular method used in any given knowledge base is selected by the programmer. Of the methods provided two were deemed too simplistic to be used in this project. The first of these two methods is a simple true or false method. In this method, once a rule assigns a value of true or false to a choice the choice is 100% true or false regardless of the content of any other rules.

In the second method the rules assign to the choices probabilities of 0 to 10, and then the probability accounting function averages the totals received by each choice. The average value, however, is overridden by rules that assign values of 0 or 10 to a choice. In such cases the choice receives a value of 0 or 10 regardless of what the average value may have been. In the event that a choice is assigned both a 0 and a 10 the value that was assigned first takes precedence and becomes the final value that is assigned to that choice.

The third method of combining probabilities is based on assigning values of from -100 to +100 and averaging all the values received. Rules that assign values of -100 and +100 do not lock the probability values to either of those levels.

The fourth and fifth methods are similar in that they are based on a probability value scale of 0 to 100. In the rules these values appear as fractions of 100, i.e. 0/100 to 100/100, and as with the other methods the rules assign values to the choices and the probability accounting method combines the values assigned. According to the EXSYS manual one of these methods combines the assigned values as independent probabilities (equation 1) and the other as dependent probabilities (equation 2).

$$NT = 100 * \{1 - [(1 - OT) * (1 - WA)]\} \tag{1}$$

where:

NT = new total (percent)
OT = old total (decimal)
WA = weight assigned by new rule (decimal)

$$NT = OT * WA \tag{2}$$

where:

NT = new total (percent)
OT = old total (percent)
WA = weight assigned by new rule (decimal)

Although the EXSYS manual provides the equations used in these two methods, the overall explanation of the methods is not clear and leaves some doubt as to what is meant by dependent and independent. The question that arises is best explained by referring to a knowledge base with two choices and two rules. Does dependent mean that given that choice 1 has a final probability of 99%, choice 2 can only have a probability of 1%, or does it mean that given that rule 1 assigns a probability of 99% to choice 1, rule two should only assign a probability of 1% to choice 1?

In an attempt to determine the correct application for each of these two probability methods, test were done on a controlled group of EXSYS knowledge bases. The tests

consisted of running a total of 12 EXSYS runs with six different knowledge bases. Each knowledge base was run in both the dependent and independent mode. The differences between each of the six knowledge bases were closely controlled, making sure to vary only one parameter from one knowledge base to the next. The total number of rules, number of true rules, and number of choices were the parameters that were changed.

The tests showed that there is no dependence between choices, and thus it is possible in either mode to have more than one choice with 100% probability assigned. The tests further showed that both the dependent and independent methods consider only the true rules. That is, that the final probabilities are the same for a given set of inputs regardless of the number of non-true rules that are in the knowledge base. This means that it is not necessary to plan the rules such that the combined probabilities assigned to any given choice equal 100%.

The results of the tests clearly showed what the terms "*independent*" and "*dependent*" do not refer to; however, the question of how to use each method remained unanswered. When statistical text books failed to yield any further assistance the EXSYS technical support department was contacted for help.

Mr. Mike Summers of Exsys Incorporated explained that the two methods in question were originally developed for one of the earliest successful expert system experiments, a medical diagnosis expert system called Mycin (see Van Horn, 1986). According to Mr. Summers, the two methods are not based on formal statistical probabilities but are "based on the model that the Mycin researchers developed to represent the reasoning methods of the medical experts" whose knowledge was programmed into Mycin (Summers 1988).

Mr. Summers explanation of how best to use the two methods is paraphrased as follows:

> 1. The independent method implies that the weight assigned to a choice by a rule is independent of the weight assigned to the same choice by any other rule. In this method the first true rule that assigns a weight to

a choice gives to that choice the total weight assigned. After the first rule, all additional true rules that assign a weight to the choice add to the choice's existing total, or the rule's assigned value times the difference between 100% and the choice's current value. Equation (3) describes this process better than equation (1), and it can be shown that equations (1) and (3) are equal. This method is highly recommended for use in knowledge bases that are designed to determine the choices that apply as the solution to a problem, thus it applies to all of the knowledge bases in this expert system.

$$NT = OT + [(100 - OT) * WA] \tag{3}$$

where:

NT = new total (percent)
OT = old total (percent)
WA = weight assigned by new rule (decimal)

2. The dependent method implies that the weight assigned by any true rule depends on the weight that is assigned by all other true rules. As in the independent method, the first true rule that assigns a weight to a choice gives to that choice the total weight assigned. After the first true rule, all other true rules that assign a weight to that choice change the value of the choice's existing total to the product of the existing total and the weight assigned by that rule. This method is only recommended for knowledge bases that attempt to reduce a set of possible solutions by eliminating all solutions that do not apply.

Based on Mr. Summers' explanation the independent method for combining probabilities was selected for use in the knowledge bases of this expert system.

Chaining and Logic

The purpose of this section is to explore how the EXSYS expert system shell implements chaining and to look at important lessons learned through working with EXSYS.

The normal EXSYS chaining mode is backward chaining, but with the addition of built-in commands to the EXSYS.CFG file EXSYS can be made to work in a forward chaining mode and in two other modes that are distinct combinations of forward and backward chaining. Since only backward chaining has been used in this project, the remainder of this section will deal only with backward chaining.

Under the normal backward chaining scheme, EXSYS rules are not tested unless the tail of the rules (THEN or ELSE) contain one of the following:

1. A choice being assigned a probability.

2. A variable to be displayed being assigned a value.

3. A variable or qualifier value assignment needed to test the head of another rule that must be tested because its tail contains one of the above two conditions.

Note that the third of the above conditions embodies the strength of the backward chaining algorithm because it reduces the number of inputs that the user must supply. When EXSYS, in the normal backward chaining mode, needs a value for a variable or qualifier it first looks to see if it can derive the value by backward chaining to other rules. If it cannot, EXSYS next looks to see if the needed value can be supplied by an external program such as the data access and computation programs of this expert system. Finally, if the data cannot be obtained from an external program, the user is asked to make a data entry. Thus EXSYS attempts to minimize the number of inputs the user must provide by making user inputs the last available method of data acquisition.

In general the EXSYS backward chaining mode performs as described in the EXSYS manual. One peculiar characteristic of the backward chaining mode whose use is not explained in the manual is that the developer has the option to program the number of rules that should be used to derive data. The EXSYS editor presents this option as follows:

Number of rules to use in data derivation

1. Attempt to apply all possible rules.
2. Stop after first successful rule.

Results of tests conducted to gain an insight into this feature showed that with the first option EXSYS will backward chain to all rules that can possibly be used to derive the datum it is trying to find and keep the value assigned by the last rule that is tested. When the second of the two options is used, the first and only the first applicable rule tested will be used for data derivation. Thus, since one option keeps the results of the last rule tested and the other the results of the first rule tested this feature can be used to determine if

conflicting rules exist in a knowledge base. In a knowledge base free of conflicting rules the results will be independent of the data derivation option selected.

A further look into the use of these options indicates that only the second of the two options should be used in a finished knowledge base. First, there is the element of time. Since the second option stops back-chaining for a given datum once a value has been found, execution is faster than it is with the first option. Second, and most importantly, though it will produce the same results the second option will obscure the logic of a knowledge base. If unneeded rules are tested after the needed datum has been derived, unneeded inputs may be requested from the user. If unneeded questions are asked, and the user exercises the option to ask "WHY" the expert system needs an input, the logic displayed by the expert system will not make sense because the needed datum has been found will be displayed as will the fact that the expert system is still trying to find (or find again) that datum.

Calling External Programs

EXSYS's ability to interface with external programs is a very useful tool that has been essential to the success of this project. In general, EXSYS provides four different methods of interfacing with external programs. These methods, all of which have been tested during this project, are as follows:

1. Calling programs at the start of an EXSYS run.
2. Calling programs from variables.
3. Calling programs from the tail of rules.
4. Calling programs from the report generator.

Only the second method listed is currently being used in the knowledge bases of this expert system; however, the usefulness of all four methods in HVAC design expert systems has been recognized and is noted below.

In the first method listed, EXSYS can call one program at the start of an EXSYS run. EXSYS loads the rules of the knowledge base and before reading any data file or asking any questions calls a designated external program. Once the external program's execution ends, EXSYS reads any data that is returned by the program, reads the REPORT.DAT data file and begins its consultation. Since calling the program is the first thing that EXSYS does it is not possible for EXSYS to pass any data to the external program, therefore the external program must get all of its data from the user or from a data file that was prepared before EXSYS is started. As noted the external program may return data to EXSYS although it need not do so.

A possible HVAC application of this method could be to call a cooling and heating load calculation program at the start of the system selection knowledge base. The user would then enter into the load calculation program all the data needed by the program and the program would return the resulting loads to EXSYS for use in system and equipment selection.

As noted, the second method, calling an external program from a variable, is the method that has been used throughout this project. According to the EXSYS manual, calling an external program from a qualifier is also possible and is essentially the same as calling a program from a variable.

The usefulness of this method, and the reason that it has been used while the other three methods have not, lies in the fact that of all the external call methods this is the only method that is capable of *passing and receiving* data to and from the external program. The method is useful, but caution must be exercised when programing with this interface method.

First, the external program must be capable of using the input and output files designated by the EXSYS configuration file (EXSYS.CFG). For this expert system the files are OUTPUT.DAT for passing data to the external program and INPUT.DAT for receiving data from the external program. The format in which the external program

writes data to the INPUT.DAT file must be as shown in Figure 8. An additional input/output option that can be used with this interface method is that if the external program can receive command line inputs, EXSYS is capable of passing data to the program in the command line. Since this second method for calling external programs has been used extensively in this expert system, an explanation of its syntax is called for.

Recall that EXSYS variables have both a name, i.e. [D-B TEMPERATURE] and a variable description, i.e. "the interior design dry-bulb temperature in degrees F." The syntax of the second interface method requires the addition of a program run statement at the beginning of the variable description. For example:

> RUN(GETDATA2 SUMMER.DAT [LOCATION] V2 V3 /C /M) the interior design dry-bulb temperature in degrees F

The general form of the "RUN" statement follows:

RUN(filename data labels options)

In this syntax, "filename" is the name of the executable program file and may be entered without the file extension. GETDATA2 is the executable file name in the above example.

If the external program is to receive data from EXSYS, the data follows the executable file name. This is represented by the word "data" shown in the general form of the syntax and may consist of numeric constants, string constants, numeric variables, string variables, or any combination thereof. The data can be passed either on the command line or in the OUTPUT.DAT file depending on the external program's input capabilities. The method used to designate which data passing option (command line or OUTPUT.DAT) is used will be discussed later.

One caution that must be noted when passing data on the command line is that any variables that are being passed must have a value assigned to them before the external program is called. This is not noted in the EXSYS manual and the results of violating this rule are very interesting. Briefly, if a variable without a value is passed to an external

program, EXSYS will ask for the input to the variable and then call the external program, but it will not pass the newly acquired data to the external program. Thus the variable must have a value assigned to it *before* the external program is called.

In the above example "SUMMER.DAT" and [LOCATION] are the data that are being passed to GETDATA2. Note that "SUMMER.DAT" is a string constant and [LOCATION] is a string variable.

In order for an external program to pass data back to EXSYS in the format shown in Figure 8, either the program must have the variable labels (i.e. V1, V2, etc.) programmed into it or EXSYS must pass the labels to the program as data. The approach that has been used in this project is the latter, so the labels follow the data in the general form of the "RUN" statement. Note that as far as EXSYS is concerned the labels are data (string constants). In the above example "V2" and "V3" are the variable labels.

The reason that passing labels as data is preferred over programing labels into the external programs is that by passing labels as data the same program, for example GETDATA1 and GETDATA2, can be used to pass data to any number of variables. The unacceptable alternative would be to have a "GETDATA" type program for every group of variables that needs data from external data base files.

Regardless of the method used to get labels into the external program extreme caution must be used anytime changes are made to EXSYS variables. Labels for EXSYS variables correspond to the order in which variables are added to a knowledge base. For example, the first variable added is V1 and the tenth is V10; however, if after the tenth variable is entered one of the variables before it (i.e. V1 thru V9) is removed, V10 becomes V9 and any program still sending data with a V10 label is in error.

After the last variable label in the "RUN" statements come the "options". There are several valid options found in the EXSYS manual but only /C and /M pertain to this project.

The default option for EXSYS to pass data to an external program is for EXSYS to use the OUTPUT.DAT file. The /C in the "RUN" statement overrides the default and tells EXSYS to pass data in the command line.

When EXSYS calls an external program, as a default it expects to receive a single value for the variable that is associated with the external program. If more than one value is returned by the external program an error will occur unless the /M option has been included in the "RUN" statement. Thus the /M tells EXSYS to expect more than one data value. In the above example the selected options are /C for command line data passing and /M for multiple data values returned to EXSYS.

Since EXSYS attempts to back chain to find a value for variables before trying the external program associated with the variable, it is imperative that variables that call programs not have a back chaining solution. If EXSYS finds a way to assign a value to such a variable the external program will never be called.

In the third method of calling external programs a "THEN" or "ELSE" statement in an EXSYS rule calls an external program, the program runs, and then, once the program finishes running, control returns to EXSYS and the consultation with the selected knowledge base continues as usual. With this method it is possible for EXSYS to pass data to the external program; however, it is not possible for the program to return any data to EXSYS. This is because EXSYS is programmed not to expect data from this type of external program call.

At first thought this method may seem fruitless; however, it is a method ideally suited for building front-end programs. The term front-end program was briefly noted in chapter I with reference to a paper by Liu and Kelly (Liu and Kelly 1988). Briefly, front-end programs, which need not be expert systems, are user friendly programs used to collect inputs for complex, and often hard to use, simulation programs. If the front-end program is to do more than just ask the user for inputs and send the inputs to the simulation program, it can be developed using an expert system shell. In such a case the

expert system would be used to assemble the user inputs as well as to make decisions such as determining the proper default values needed to run the simulation program. Since the front-end program only needs to send data to the simulation program, it is possible to use this particular type of EXSYS external program interface when building an expert system for a front-end program.

The last of the external program interface methods provided by EXSYS is also well suited for use in a front-end program. The only difference between this method and the previous methods is the syntax used to call the external programs. With this method an EXSYS run takes place, then just before the EXSYS results are displayed, EXSYS writes data to the OUTPUT.DAT file for use in the external program. EXSYS then calls the external program, and when the external program finishes its run control is returned to EXSYS and the EXSYS results are displayed.

Recursion

Under normal circumstances EXSYS looks at a rule only once. If the head of a rule is true EXSYS executes the rule and assigns values for variables, qualifiers or choices as indicated by the tail of the rule. If the head of the rule is false EXSYS ignores the rest of the rule. Once a rule is proved false it is not looked at again during the consultation.

During the development of this expert system the need arose for testing certain rules more than once. This need occurred both for cases in which a rule had been proven true and in those in which the rule had been proven false. An example of this is as follows: The preliminary design knowledge base selects several system types, six of which are passed by UPDATE to the preliminary cost knowledge base. These systems are passed in rank order from most to least probable according to the assigned probabilities. The system types are assigned to six variables named [RECOM SYST 1] through [RECOM SYST 6]. The function of the preliminary cost knowledge base is to

then determine a cost for each of the six systems and assign the cost to the variables [SYST 1 COST] through [SYST 6 COST].

Since there are eleven possible systems that the preliminary design knowledge base can select, a brute force method of completing this task would be to write at least eleven rules for each of the six recommended system variables:

RULE 1: IF: (1) [RECOM SYST 1] = Built up VAV
THEN: (1) [SYST 1 COST] = ...

RULE 2: IF: (1) [RECOM SYST 1] = Packaged VAV
THEN: (1) [SYST 1 COST] = ...

.
.
.

RULE 11: IF: (1) [RECOM SYST 1] = Single Zone
THEN: (1) [SYST 1 COST] = ...

RULE 12: IF: (1) [RECOM SYST 2] = Built up VAV
THEN: (1) [SYST 2 COST] = ...

RULE 13: IF: (1) [RECOM SYST 2] = Packaged VAV
THEN: (1) [SYST 2 COST] = ...

.
.
.

RULE 22: IF: (1) [RECOM SYST 2] = Single Zone
THEN: (1) [SYST 2 COST] = ...

Now we would write eleven identical rules using [RECOM SYST 3] and so on for a total of 66 rules. However, since in most cases the cost estimate is dependent upon other variables such as building type and locations, it is necessary to write more than one rule per system type. Thus the number of rules would quickly jump from eleven per variable.

Instead of this brute force approach, the method that was used in this project was to begin with eleven rules that assign the value of the systems to a temporary variable:

RULE 1: IF: (1) [RECOM SYST 1] <> "NONE"
THEN: (1) [CURRENT SYSTEM] = [RECOM SYST 1]
and (2) [SYST 1 PRICE] = [CURRENT SYST COST]

RULE 2: IF: (1) [RECOM SYST 2] <> "NONE"
THEN: (1) [CURRENT SYSTEM] = [RECOM SYST 2]
and (2) [SYST 2 PRICE] = [CURRENT SYST COST]

.
.
.

RULE 11: IF: (1) [RECOM SYST 6] <> "NONE"
THEN: (1) [CURRENT SYSTEM] = [RECOM SYST 6]
and (2) [SYST 6 PRICE] = [CURRENT SYST COST]

With these rules it was then possible to write a set of rules for each of the eleven possible systems using the [CURRENT SYSTEM] and [CURRENT SYST COST] variables instead of the individual system and system cost variables. Assume, as was done in the brute-force method, that only eleven rules are needed to estimate the price of the systems. All that was then needed was eleven more rules. Thus with this method 22 rules can do the work of 66 rules using the brute-force method:

RULE 12: IF: (1) [CURRENT SYSTEM] = Built up VAV
THEN: (1) [CURRENT SYST COST] = ...

RULE 13: IF: (1) [CURRENT SYSTEM] = Packaged VAV
THEN: (1) [CURRENT SYST COST] = ...

.
.
.

RULE 22: IF: (1) [CURRENT SYSTEM] = Single Zone
THEN: (1) [CURRENT SYST COST] = ...

Note that in this scheme rules 1 through 11 assign, one at a time, the values of all the recommended systems to the [CURRENT SYSTEM] variable and that they also assign to the appropriate [SYST # COST] variables the value of the [CURRENT SYST COST] variable. Rules 12 through 22, on the other hand, estimate the values for the [CURRENT SYST COST] variable.

The problem with this scheme as explained thus far is that if [RECOM SYST 1] has the value "Single Zone," then when rule 1 assigns the value "Single Zone" to the [CURRENT SYSTEM] variable rules 12 through 21 will be found to be false since (in this example) only rule 22 applies to single zone systems. Thus when rule 2 changes the value of [CURRENT SYSTEM] to another system all rules 12 through 21 that could possibly apply have already been found to be false and EXSYS cannot find an answer.

Fortunately EXSYS provides a statement called "CLEAR" which makes it possible to reuse rules that have been found to be true or false. To make a rule reusable when it has been found to be false we place the "CLEAR" statement in the ELSE part of the rule. Thus, to make the above scheme work, rules 12 through 22 are modified as follows:

```
RULE 12: IF:   (1) [CURRENT SYSTEM] = Built up VAV
       THEN:   (1) [CURRENT SYST COST] = ...
       ELSE:   (1) CLEAR(R 12)

RULE 13: IF:   (1) [CURRENT SYSTEM] = Packaged VAV
       THEN:   (1) [CURRENT SYST COST] = ...
       ELSE:   (1) CLEAR(R13)
                .
                .
                .
RULE 22: IF:   (1) [CURRENT SYSTEM] = Single Zone
       THEN:   (1) [CURRENT SYST COST] = ...
       ELSE:   (1) CLEAR(R 22)
```

If the "CLEAR" statement is placed in the THEN part of a rule, it clears the rule when it is true. Similarly, the CLEAR statement can also be used to remove the values from variables after they have been used.

Summary

In summary, the process of familiarization with the EXSYS program consisted of a tutorial approach to the EXSYS manual. Two small knowledge bases (about 50 rules each) and several very small (about 10 rules each) test knowledge bases were built and scrutinized to observe the behavior of the expert system shell. The significance of this process is that:

1. It led to the design and completion of all the programs that interface with EXSYS: GETDATA1, GETDATA2, the basic form of all the computation programs, HVAC, PROPRINT, and most importantly, UPDATE.

2. It proved the feasibility of the HVAC expert system structure illustrated in Figure 3.

3. It clarified information not thoroughly explained in the EXSYS manual: actual syntax of qualifiers, method for working around the 30

qualifier limit, proper use of the probability methods, selection of number of rules used for data derivation, problems with calling external program from variables, and use of the CLEAR statement.

CHAPTER V

HVAC DESIGN KNOWLEDGE BASES

Overview

The knowledge programing for an expert system is made up of two distinctly different processes. These are determining what knowledge to program and programming that knowledge. In this project the former proved the most difficult of the two tasks and thus required careful consideration. Determining what to program includes both selecting the proper topics to program and searching out the expert's knowledge about the selected topics.

Selection of Knowledge Domains

The literature provided some general guidance as to what type of knowledge is well suited for programming into expert systems. Townsend and Feucht (1986) explain that knowledge domains ideally suited for programming in knowledge based expert systems have the following characteristics:

> "1. The data and knowledge needed to program the domain are reliable and should not change with time.
>
> 2. The domain of possible solutions is relatively small.
>
> 3. There is at least one acknowledged expert capable of explaining his knowledge and the methods used to apply knowledge to the problem.
>
> 4. The problem solution involves formal reasoning. If the solution involves a procedural analysis, a traditional computer program is better suited."

The first of the above characteristics is of little concern in this project. While HVAC design knowledge certainly does change, the changes take place over long periods

of time (one or more years) making the domain static enough to make it suitable. Changes are slow enough that they can be incorporated into program revisions.

The second characteristic points to small knowledge domains and strongly supports the HVAC design expert system structure explained in Chapter III. Recall that the main intent of the structure is to separate the knowledge domain into several relatively small knowledge bases to save computer memory. These smaller knowledge bases and the resulting smaller knowledge domains help this expert system meet the second desirable characteristic of knowledge domains because they divide the whole into smaller groups.

Since there are many acknowledged HVAC experts the third of Townsend and Feucht's characteristic is of no concern here. However, because of the procedural nature of HVAC design the last characteristic requires much attention. The need for concern here can be understood if one surveys HVAC design text books and the content of HVAC design courses. In such a survey one finds that the majority of the topics covered are procedural and thus are not the type of knowledge best suited for programming in knowledge-based expert systems. Examples of such procedural topics are load calculations, energy estimating, and duct and pipe sizing. Keeping in mind that the knowledge to be programmed should involve formal reasoning, the problem of determining what knowledge to program is one of finding useful HVAC design knowledge that meets the *formal reasoning* requirement. Two vital sources of information that helped to solve this problem were a survey questionnaire sent to a group of novice engineers and classroom interaction with mechanical engineers who are currently involved in HVAC design.

HVAC Design Survey

The purpose for the survey was to gain some idea of what type of knowledge novice mechanical engineers would find useful in an HVAC design expert system. The

survey questionnaire (found in Appendix B) was mailed to 120 United States Air Force (USAF) mechanical engineers working in the facilities engineering field. Personal background questions in Section I of the questionnaire were used to separate the respondents by experience level. This was done to isolate the responses of the target group, which included graduate mechanical engineers with less than four years of HVAC design experience but with at least some HVAC design experience and/or HVAC design training.

Of the 120 surveys that were mailed out 92 were returned. Seventy-five of the respondents had less than four years of HVAC design experience and were involved in HVAC design either by actual work experience or by attendance in a post-graduate HVAC design course or both. It is the surveys returned by these 75 respondents that were used to apply the survey results to the problem of determining the type of knowledge needed in the expert system. The survey results for all respondents are found in Appendix C.

The HVAC design questions in the survey questionnaire were separated into two sections. Section II was designed to determine the type and size of HVAC systems that the respondents were called upon to design most often. Table 3 contains the results of this section. Note that the results shown on Table 3 are based on only 68 respondents. These are the 68 respondents with actual HVAC design experience of four years or less. Table 4 contains the results of section III of the survey. The questions in section III were designed to determine the aspects of HVAC design that the respondents felt they needed most help with. The results in Table 4 are based on the complete group of 75 respondents with four years or less of HVAC design experience and/or attendance at a HVAC design short course.

The usefulness of the survey was limited by the fact that many of the design topics in section III of the survey are the same procedural (computational) type topics, i.e. load calculations etc., that were noted above as not being appropriate for expert systems. Nevertheless, the survey did provide a good starting point for the knowledge selection.

Table 3

Results of survey section II for the 68 respondents with less than four years of HVAC design experience.

"A" FREQUENTLY USED TO "C" INFREQUENTLY USED				
ITEM	SYSTEM DESCRIPTION	A	B	C
HEATING				
4	Gas/oil furnaces	16	53	31
5	Gas radiant heaters	9	31	60
6	Unit heaters	22	63	15
7	Gas/oil fired boilers	22	57	20
8	Electric resistance heat	4	63	32
9	Heat pumps	10	53	37
REFRIGERATION				
10	Reciprocating	29	54	16
11	Centrifugal	6	47	47
12	Absorption	0	18	82
HEAT REJECTION				
13	Air cooled	47	43	10
14	Water cooled	16	46	38
15	Evaporative	7	38	54
16	Cooling tower	13	54	32
COOLING				
17	Packaged DX	41	47	12
18	Packaged absorption	1	9	90
19	Heat pump	13	47	40
20	Built-up DX	13	50	37
21	Chilled water	41	50	9
22	Evaporative Cooler	13	31	56
PIPING SYSTEMS				
23	Four pipe	9	51	40
24	Three pipe	6	40	54
25	Two pipe	40	53	7

Table 3, continued.

"A" FREQUENTLY USED TO "C" INFREQUENTLY USED				
ITEM	SYSTEM DESCRIPTION	A	B	C
AIR SUPPLY/DISTRIBUTION				
26	Single zone	60	35	4
27	Single duct with reheat	16	47	37
28	Single duct VAV	6	48	46
29	Dual duct VAV	1	18	81
30	Multi zone	37	47	16
31	Fan coil units	35	57	7
VENTILATION SYSTEMS				
32	Commercial kitchen exhaust	10	56	34
33	Industrial exhaust	10	56	34
SCOPE OF SYSTEMS				
34	Fractional to 20 tons	54	46	0
35	20 to 100 tons	21	60	19
36	Greater than 100 tons	6	22	72
SCOPE OF BUILDINGS				
37	Less than 5,000 SF	50	43	7
38	5,000 to 10,000 SF	37	56	7
39	10,000 to 50,000 SF	19	65	16
40	Single story	63	34	3
41	Two story	16	53	31
42	Three to four story	6	24	70
43	Five or more stories	0	9	91

Table 4

Results of survey section III for the 75 respondents with less than four years of HVAC design experience/and or who have attended a HVAC design short course.

"A" HELP NOT NEEDED TO "D" HELP NEEDED"					
ITEM	DESCRIPTION OF TOPIC	A	B	C	D
44	Cost estimating	57	31	11	1
45	Energy use estimating	27	56	13	4
46	Life cycle costing	15	53	25	7
47	Specifications	41	40	13	5
48	Construction details	11	37	40	12
49	Maintenance details	16	32	35	17
50	Psychrometrics	41	52	5	1
51	Ventilation / infiltration	43	52	4	1
52	Heating / cooling loads	48	47	5	0
53	System selection	21	40	32	7
54	Equipment selection	20	36	33	11
55	Equipment noise control	5	31	40	24
56	Air distribution noise control	20	36	33	11
57	Air distribution design	28	45	19	8
58	Duct design / fan selection	35	50	11	4
59	Piping design / pump selection	33	45	20	1
60	Control system design	12	27	31	29

The result was that the general topics to be programmed were selected with the help of both the survey and the HVAC design process outline found in Section 15 of the expert system specification (Appendix A). This resulted in the knowledge base structure which is shown in Figure 9.

Figure 9 shows the knowledge bases in the same hierarchy in which they appear in the system design menu of the HVAC program. The arrangement of the knowledge bases is modeled on the project design chronology familiar to the researcher and outlined by Mueller and Associates (1986).

HVAC Design Course

In addition to the survey, feedback from students attending HVAC design and HVAC controls design courses at the Air Force Institute of Technology (AFIT) provided useful inputs to the selection of appropriate knowledge domains for the expert system. Each of these courses is offered three times per year with 20 to 30 students enrolled in each offering.

Since one of the main reasons for the expert system is to provide design help and training for inexperienced HVAC engineers, the views of students attending these courses were very appropriate for this project. The interaction with these students was never specifically directed at eliciting information for this project; however, many of the comments from these students provided useful information for this research.

Through interaction with the students it was noted first that topics that had been considered too trivial for the expert system are not trivial to inexperienced engineers. For example, the selection of required equipment, originally thought to be trivial, is not considered trivial by many of the students. The reference here is to generic equipment type selection, i.e. a VAV system needs a central air-handling unit and VAV boxes, and not to specific equipment selection from catalogs. Second, it was also found that within the procedural topics of HVAC design exist subjective subtopics that require reasoning

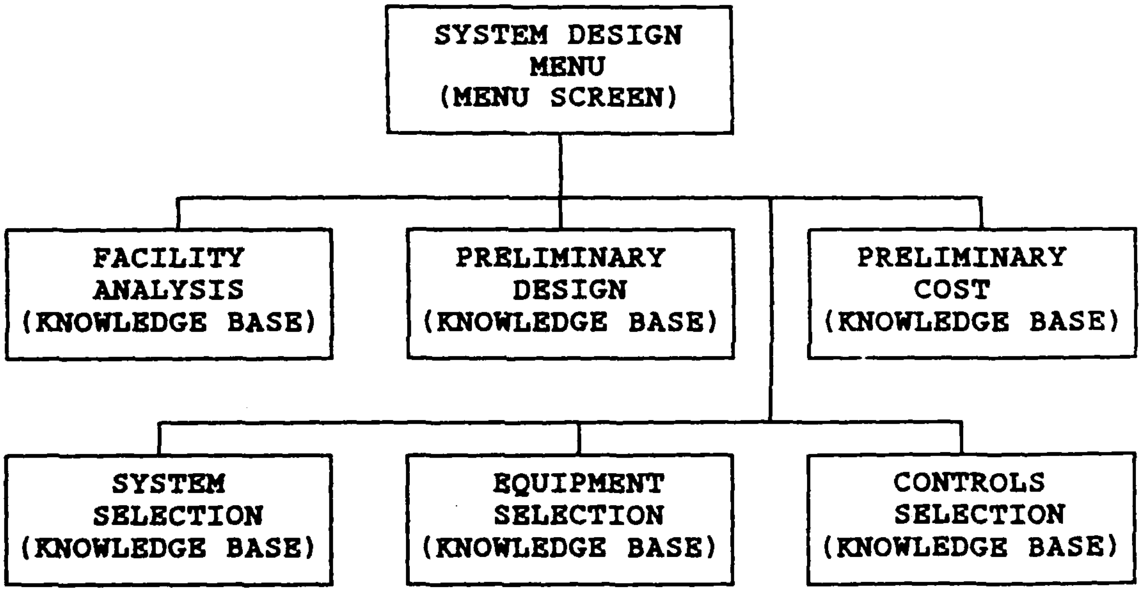

Figure 9

Knowledge-base structure.

and are therefore appropriate for programming in an expert system. An example of this type of knowledge is the topic of heating load adjustment for intermittent system operation warm-up.

The interaction with several groups of these students led to a well defined list of topics that could be programmed into the expert system. These are presented in Table 5. More importantly, however, the AFIT students provided an excellent forum for judging the needs of inexperienced engineers. Their feedback, and the feedback of similar groups of students, are perhaps the best possible source of guidance for this and future HVAC design expert systems. Their limited experience and knowledge enabled them to see things that more experienced engineers overlook due to their familiarity with the subject.

Knowledge Acquisition

The next step after the selection of the proper topics for the expert system was to acquire the knowledge (i.e. the thought process) that was to be programmed. It should be noted that the processes of knowledge acquisition and knowledge programming occurred concurrently in an interactive manner. In fact the two processes, though clearly different, are mutually dependent processes that cannot be effectively accomplished if they are carried out separately. The purpose of the present section is to present the alternatives encountered and used for knowledge acquisition.

In the literature strong emphasis is placed on the concept of the knowledge engineer interviewing the domain expert ("knowledge engineer" meaning an expert in the programming of knowledge-based expert systems) (Brothers 1988; Kosten and Maher 1986; Townsend and Feucht 1986 and Van Horn 1986). In this scenario the knowledge engineer conducts interviews, programs the knowledge he has gained through the interviews and validates the resulting knowledge base by allowing the domain expert to review the resulting rules. Brothers (1988) describes this process in detail. A second

Table 5

Topics for knowledge bases.

KNOWLEDGE BASE	TOPICS PROGRAMMED
Facility Analysis	Verification of building type Verification of heat/cool requirements Selection of outside design conditions Selection of inside design conditions Building envelope evaluation
Preliminary Design	Estimate of heating and cooling loads Initial system selection
Preliminarv Cost	Initial cost estimate for selected systems Estimate of mechanical room space for selected systems Estimate of economic life for selected systems
System selection	Selection of HVAC systems
Equipment selection	Selection of equipment for selected systems
Controls selection	Selection of controls for selected systems

method of knowledge acquisition, called expert system construction on a "cerebral basis," occurs when the knowledge programming is done by the domain expert (Brothers 1988).

The advantage to the first method, it would appear, is that the competent programming expert (knowledge engineer), if capable of drawing out the knowledge from the domain expert, could better program the knowledge than could the domain expert. This is based on the often true assumption that the domain expert is not a competent knowledge engineer himself.

Alternately, the advantage to the second method is that the domain expert, being much more aware than a knowledge engineer of what direction the expert system should take, is better capable of guiding the development. The domain expert may not, by lack of programming knowledge, be capable of taking full advantage of all the nuances of expert system programming. However, his clear view of the topic to be programmed could offset his lack of programming expertize and produce a better product than could be possible by the first method.

Each method has its advantages and this project does not provide the means for a valid, objective comparison of the two methods. Since the researcher is more experienced in HVAC design than he is in knowledge engineering it can be said that this project was conducted using the cerebral basis method. However, in order to enhance the final product, several HVAC design knowledge sources other than the researcher's own knowledge were used. Thus it can be said that this project used elements of both the cerebral basis and knowledge engineering methods and that the actual method used can best be classified as a hybrid.

The HVAC design experts that were interviewed for this project were Professor D. C. Hittle of the Purdue University School of Mechanical Engineering, and Captains J. A. Hudson, R. L. Boyce, and B. A. Flake, all Mechanical Engineering Professors at the Air Force Institute of Technology.

An added element in this project is that for knowledge acquisition much use was made of HVAC design manuals (ASHRAE 1985; ASHRAE 1987; Carrier 1965; and Stanford 1988). These manuals were used for both *knowledge acquisition* and data gathering, while other manuals were used solely for data gathering. A United States Air Force construction regulation was also used as a knowledge source for the applicable cases where the law requires adherence to the regulation's HVAC design requirements (USAF 1986).

As no precedent was found in the literature for the use of published data for knowledge acquisition, it is possible that doing so is not an acceptable practice in expert system knowledge programming. Nevertheless, the wide acceptance enjoyed by ASHRAE as the research leader in the HVAC industry justifies the use of ASHRAE literature as a source of knowledge. It would be difficult to find a HVAC expert whose knowledge of the subject is not influenced by ASHRAE research.

Programming of the Knowledge Bases

The purpose of this section is to explain the six knowledge bases that make up this expert system. Here the knowledge bases are described in terms of their purpose, their inputs and outputs and their technical content. The complete listings of the knowledge bases are found in appendices G through L of Herrick Laboratory Report HL 89-37 (Camejo and Hittle 1989).

Facility Analysis Knowledge Base

Purpose. As Table 5 shows, several different topics are programmed into this knowledge base. The overall purpose of this knowledge base is twofold. First, it determines information that is needed for the other knowledge bases to do their work, and second, it determines some useful information for the user to do his work.

In each case the idea is the same; namely, that before beginning a design, certain information has to be gathered or determined. The information that is determined for the

other knowledge bases is the type of building, the project location, the need for heating or cooling or both, winter interior and exterior design temperatures and certain characteristics derived from the type of building, i.e. high internal loads, high latent loads etc.

While all the information determined for use in the rest of the expert system is of use to the user, the knowledge base determines some other parameters that are not used by the other knowledge bases. These include all interior and exterior design conditions and weather data, and the ASHRAE Standard: Energy Conservation in New Building Design (ASHRAE 1980) evaluation of the building envelope. These are extremely important to the designer but not to the other knowledge bases of the expert system.

Programmed Knowledge. The first function of this knowledge base is to determine the project location. This is done by calling the GETDATA1 program to access the project location code for a variable called [LOCATION]. The only input from the user is the name of the project location. The user selects the location from a menu and GETDATA1 returns to EXSYS the location code, latitude and longitude coordinates and elevation above mean sea level of the location. With this information known the knowledge base can use the GETDATA2 program to access the winter and summer weather data files and obtain the weather data for the project location. The weather data, compiled from ASHRAE (1972 and 1985) and U. S. Department of Commerce (1986), is very complete and includes all the usual summer and winter design temperatures plus the median of annual winter extremes, and winter wet bulb temperatures, summer average daily range, and annual cooling and heating degree days. Thus this function performs like a smart data base, providing the user with data for load and psychrometric calculations.

Next the knowledge base performs a function for which the need was never foreseen. When the user enters a building type from the expert system menu, this knowledge base uses approximately 40 rules to ensure the user entered the correct building type for the project. This is needed because certain building types, although

similar to the user, are very different to the knowledge bases of the expert system. For example, if the user enters "motel" for a luxury hotel the expert system would be steered in the wrong direction when it completes its load estimate and system selection processes. Thus when the user enters motel the expert system wants to know the number of floors, since few motels are more than three stories tall. The expert system may also ask the user to verify if the building is a hotel or motel in the event that the building is three stories high.

Once the building type is verified the knowledge base begins to work on recommending the design conditions for the building and on determining the need for cooling and/or heating. Determination of the need for heating or cooling is a function that, like verification of building type, appears trivial. This is because under normal circumstances the engineer, familiar with the common practices in his region, knows exactly when to heat, when to cool and when to do both. In most cases it is a decision completed without thought. A simple way of completing this task is to have the knowledge base ask the user what common practice is in the project region. This, however, is wrong for two reasons. First the user may not know the answer, and second the expert system must be as self-sufficient as possible, equally capable of dealing with both complex and trivial domains.

In the first case, many HVAC systems designed by experienced engineers unfamiliar with the project region suffer from poor designs. For example, cooling systems designed by companies in the Northeast for the southeastern U. S. and Central American climates are notorious for their inability to dehumidify even during peak load conditions. Thus it is not advisable to depend too much on inexperienced engineers.

In the second case, quantifying heuristic knowledge is one of the goals of this project, and on the surface few topics could be more intuitive then the decision of whether to heat or cool. This seemingly trivial decision requires some creativity to program.

however, because the decision normally does not require any conscious thought process that can be duplicated in the knowledge base.

The approach taken to program the heat or cool decision is illustrative of the general approach that in the end was applied to most of this expert system. In this approach a group of rules are programmed near the beginning of the knowledge base. These rules are designed to assign qualifier values based on the type of building being designed and on other general characteristics of the building. The following rules illustrate this point:

```
(1)  IF:  (1) [BUILDING TYPE] = "DORMITORY"
 THEN:    (1) THE BUILDING IS NOT A CRITICAL FACILITY
   and    (2) THE BUILDING'S USE IS DOMICILIARY
   and    (3) THE MAIN PURPOSE FOR HVAC IS COMFORT
   and    (4) THE INTERNAL LOADS ARE USUALLY LOW
   and    (5) THE LATENT LOADS ARE USUALLY LOW
   and    (6) NORMAL OCCUPANCY IS 24 HOURS A DAY
   and    (7) ACTIVITY LEVEL IS OFFICE WORK and WALKING

(2)  IF:  (1) [BUILDING TYPE] = "BAR_LOUNGE"
   and    (2) THERE IS A DANCE FLOOR
 THEN:    (1) ACTIVITY LEVEL IS LIGHT EXERCISE

(3)  IF:  (1) [BUILDING TYPE] = "BAR_LOUNGE"
   and    (2) THERE IS NOT A DANCE FLOOR
 THEN:    (1) ACTIVITY LEVEL IS SEDENTARY

(4)  IF:  (1) [BUILDING TYPE] = "BAR_LOUNGE"
 THEN:    (1) THE BUILDING IS NOT A CRITICAL FACILITY
   and    (2) THE BUILDING'S USE IS ENTERTAINMENT
   and    (3) THE MAIN PURPOSE FOR HVAC IS COMFORT
   and    (4) THE INTERNAL LOADS ARE USUALLY LOW
   and    (5) THE LATENT LOADS ARE USUALLY LOW
   and    (6) NORMAL OCCUPANCY IS 24 HOURS A DAY
```

The first of these rules is an example of the general form of this group or type of rules. It assigns values to seven different qualifiers based on information that can be inferred from the type of building. The second, third and fourth rules illustrate a variation of the general concept, namely that sometimes it is necessary to use more information than just the building type. In these last three rules the existence of a dance floor is obviously

a user input. Other inputs similarly used are number of floors, type of collections in libraries and museums, and number of family units in multi-family residential buildings.

Once rules like the ones above derive general information about the project, the general information is used along with other user inputs to determine if a project requires heating or cooling or both. First there is the question of critical or non-critical facilities. Health care functions, archival storage and process air conditioning requirements all indicate critical facilities. It was decided that in all cases these must have cooling. What about heating? Heating, it was decided, would not be needed if the critical facility was located in an area where the 99% Winter design dry-bulb temperature is greater than or equal to 70^{O}F. Similar decisions were made for non-critical facilities. For example, buildings with high internal loads need cooling in all climates but do not need heating when the 99% Winter design dry-bulb temperature is above 65^{O}F, and buildings not occupied at night do not need heating if the design temperature is above 55^{O}F. Thus building use, weather, and building characteristics (i.e. high internal heat loads) are used to make decisions about heating and cooling.

In addition to making decisions based on critical or non-critical occupancy and weather, social factors were brought into the decision making. For example, regardless of location hotels and motels are cooled, even in Fairbanks Alaska. The same may be said for custom homes. Type of construction was also considered as buildings with heavy walls retain more heat than lighter buildings. Therefore, a description of the project building's construction is a required user input.

The final output from this section is one of the following four possible values given to a qualifier: cooling is needed, heating is needed, cooling and heating are needed, or only ventilation is needed. Once this decision is made the knowledge base proceeds to determine exterior and interior design conditions.

The process for determining exterior design conditions is very similar to the process for determining heating and cooling requirements. In fact, the scheme is the

same. The same general rules infer information about the project based on building type and other user inputs. Then a new group of rules determines which exterior design temperatures should be used for load calculations.

The possible results for this section are choices that recommend temperature classifications for use in load calculations. These are 1%, and 2.5% Summer design dry-bulb temperatures and 99%, 97.5%, and median of annual extremes winter design temperatures. The main decision making factor here is whether or not a building is a critical facility. Other factors are also considered; for example, the results of the previous section are used to determine the design temperature classifications, as are the expected internal loads, construction type, weather and time of occupancy. The previous section's decision of heating or cooling is crucial to avoid recommendations such as use of the 99% Winter design dry-bulb temperature for heating loads in a building the expert system has already recommended should not be heated.

A sample rule from this section is as follows:

IF:	(1)	[BUILDING TYPE] = "COMPUTER_ROOM"
and	(2)	THE PROJECT REQUIRES HEATING AND COOLING
THEN:	(1)	Use 1% Summer design D-B temperature - Probability=80/100
and	(2)	Use 97 5% Winter design D-B temperature - Probability=75/100

After the exterior design temperature classification has been selected, the knowledge base proceeds to determine interior design conditions. Like the heating or cooling decision, this is a decision that engineers often make with little or no thought; therefore, like the question of whether to heat or cool, this section was challenging because it required creating a decision process for something that is taken for granted. If asked, nearly anyone involved in HVAC design would answer that the typical interior design conditions for comfort applications are Summer 75^{O}F and 50% RH and Winter 70^{O}F. This answer would most likely be based on experience an would not be supported by a tangible decision making process.

Like the selection of exterior design conditions, the selection of interior conditions begins with prior knowledge of the knowledge base's decision on heating and or cooling. Again, one does not want to declare that a building that will not be cooled has a Summer interior design dry-bulb temperature of 75^{o}F.

For critical facilities the interior conditions are taken from the ASHRAE Handbook 1987 HVAC Systems and Applications (ASHRAE 1987). The ASHRAE recommended conditions are simply assigned to the project's interior design conditions. In these cases the user is asked to enter specific information about the use of the building; for example, types of collections in libraries and museums, and types of spaces in medical clinics, i.e. inpatient rooms, operating rooms etc. Note that inputs needed for the different sections of this and other knowledge bases are only entered once by the user.

Since the majority of HVAC applications are for providing occupant comfort, a thought process had to be developed for determining the interior design conditions of non-critical buildings. To begin with, the user is asked to enter whether the facility is privately owned or owned by the U. S. Department of Defense. This is because military facilities must adhere to certain design regulations (USAF 1986). If a facility is military, interior design conditions are determined by a few simple rules which depend on the fact that humidification is not authorized for comfort applications. Thus the Winter design condition for comfort applications in military facilities is 71^{o}F. For Summer, the interior design temperature must fall between 75^{o}F and 78^{o}F inclusive. The actual temperature is determined by subtracting 15^{o}F from the 2.5% Summer design dry-bulb temperature for the project location. If the difference exceeds 78^{o}F then the design temperature is 78^{o}F. Likewise if the difference is less than 75^{o}F the design temperature is 75^{o}F. If the difference falls between the 75^{o}F and 78^{o}F limits, the difference is the design temperature.

The summer design relative humidity, on the other hand, is the lesser of 50% RH and the relative humidity of air at conditions of 2.5% Summer design absolute humidity

ratio and the inside design dry-bulb temperature. This relative humidity value is calculated by the PSYCH1 program. The knowledge base calls the program, reads the program's output and compares it to 50% to determine the value of the design relative humidity.

The PSYCH1 program uses the 2.5% Summer design dry-bulb temperature and its mean coincident wet-bulb temperature to complete its calculations. The program uses standard psychrometric equations published in chapter six of the ASHRAE Handbook of Fundamentals (ASHRAE 1985). The equations, where applicable, were corrected in accordance with the errata that was published in the ASHRAE Refrigeration Handbook (ASHRAE 1986).

For privately owned comfort applications the military procedure was modified and applied to both Winter and Summer interior design conditions. For Summer, PSYCH1 is again used to provide information about the outside humidity in the project area. In this case PSYCH1 is used to calculate the relative humidity of air at outside absolute humidity ration and 75^{O}F. A group of rules then make predictions for inside design relative humidity based on the magnitude of this number, an assumed coil temperature of 50^{O}F, and the expected internal latent loads in the building. In the following sample rules note that no dehumidification is expected when the [CONST W SUM INT RH] is below 40%:

IF:	(1) THE BUILDING IS OWNED BY A PRIVATE CONCERN
and	(2) THE MAIN PURPOSE FOR HVAC IS COMFORT
and	(3) LATENT LOADS ARE USUALLY HIGH
and	(4) 25 <= [CONST W SUM INT RH] < 30
THEN:	(5) [INT SUMMER DESIGN RH] = 35

IF:	(1) THE BUILDING IS OWNED BY A PRIVATE CONCERN
and	(2) THE MAIN PURPOSE FOR HVAC IS COMFORT
and	(3) LATENT LOADS ARE USUALLY LOW
and	(4) 25 <= [CONST W SUM INT RH] < 30
THEN:	(5) [INT SUMMER DESIGN RH] = 30

Once the expected interior design relative humidity has been estimated, new rules in the knowledge base consider both this estimate and the expected level of activity of the building's occupants to determine a Summer interior design dry-bulb temperature.

Essentially, these new rules form a decision matrix based on the ASHRAE comfort chart (ASHRAE 1985). Three of the rules are as follow. Note that the activity qualifier and the relative humidity variable drive the changes in the selected design temperature:

IF: (1) THE BUILDING IS OWNED BY A PRIVATE CONCERN
and (2) THE MAIN PURPOSE FOR HVAC IS COMFORT
and (3) [INT SUMMER DESIGN RH] = 30
and (4) ACTIVITY LEVEL IS OFFICE WORK or WALKING
THEN: (5) [INT SUMMER DESIGN T] = 77

IF: (1) THE BUILDING IS OWNED BY A PRIVATE CONCERN
and (2) THE MAIN PURPOSE FOR HVAC IS COMFORT
and (3) [INT SUMMER DESIGN RH] = 30
and (4) ACTIVITY LEVEL IS LIGHT EXERCISE
THEN: (5) [INT SUMMER DESIGN T] = 75

IF: (1) THE BUILDING IS OWNED BY A PRIVATE CONCERN
and (2) THE MAIN PURPOSE FOR HVAC IS COMFORT
and (3) [INT SUMMER DESIGN RH] = 50
and (4) ACTIVITY LEVEL IS LIGHT EXERCISE
THEN: (5) [INT SUMMER DESIGN T] = 74

A similar set of rules is applied to the problem of determining Winter design conditions for comfort heating. In the Winter procedure the possibility of humidification is considered and recommended by the rules of the knowledge base. The measure of relative humidity is provided by the PSYCH3 program which uses the 97.5% Winter design temperature and the average exterior relative humidity for the month of January to determine the relative humidity of air at the outside humidity ratio and 70^{O}F interior temperature.

With the selection of the inside design conditions, this knowledge base has finished determining several pre-design parameters for the user of the expert system. Note that, as should be expected in an expert system, all of the items thus far determined are to some degree subjective. The decision to heat and or cool is clear cut in some instances and debatable in others, and the use of the median of annual Winter extremes as the design temperature when working with buildings of low thermal mass is accepted, rejected or debated depending on the climate and application.

The last function that this knowledge base completes is an evaluation of the project building's envelope in accordance with ASHRAE Standard 90-1980 (ASHRAE 1980). To apply this standard one classifies the building according to the standard, then, based on the classification and location of the building, one compares the thermal properties of the building's envelope to the accepted values published in the standard. This procedure is well suited for implementation in an expert system because it requires logical comparisons first to classify the building and then to see if the building's envelope meets standard.

The only additional inputs this section requires are inputs about the building's use and construction. For example, the knowledge base needs to know if the building has a crawl space, a basement, or a floor built on grade. Using new inputs and existing data the rules in the knowledge base classify the project building either as "A1," "A2" or B. These are the ASHRAE Standard 90-1980 (ASHRAE 1980) building classification codes. The "A1" and "A2" are for small residential and domiciliary buildings and the "B" is for larger residential buildings and for commercial buildings.

The procedure for applying the rest of the standard requires that the user enter values for insulation, overall heat transfer coefficients, and overall thermal transfer values for the building's walls, roof, floor, and foundation. The help file that can be displayed by the knowledge base explains these values to the user. To improve this interface it is possible to write a program that computes these values using user inputs of thermal resistances, areas, thickness, fenestration shading coefficients and thermal diffusivities.

To complete its work, the knowledge base uses the STD90-80 program to retrieve numbers from the standard. Recall that the STD90-80 program curve fits the graphs published in ASHRAE Standard 90-1980 (ASHRAE 1980). The knowledge base compares the user inputs for heat and thermal transfer coefficients to the values returned by the STD90-80 program and tells the user if the building does or does not meet the

standard. The knowledge base also displays both the user's inputs and the standard values for the user to compare.

Preliminary Design Knowledge Base

Purpose. This knowledge base and the preliminary cost knowledge base work together to quickly provide an answer to the most often asked question in HVAC design: How much will it cost to air condition this building? Experience in both civilian and military mechanical design offices indicate that engineers inevitably must put aside their present work to answer this all important question. In the civilian sector the question usually comes from a former or prospective customer who is looking at some budget figure for a project. Invariably the customer must have a price for the HVAC system within the next five minutes. In the military sector the situation is the same; however, the requirements of the military project approval process also require that the engineer estimate the total amount of floor space to be allowed for mechanical rooms. Thus, the purpose of these two knowledge bases is to provide load, cost and system space estimates from the least possible number of inputs. The intent is to use the results in pre-design budget estimates.

Programmed Knowledge. This knowledge base estimates heating and or cooling loads for the project building and makes a generic HVAC system selection based upon the loads and the application. The user can freely use this knowledge base without first using the facility analysis knowledge base. However, when this is done the user will have to make inputs that would ordinarily be provided by the facility analysis knowledge base. Some of these inputs are the scope of the project, i.e. heating, cooling or heating and cooling, the winter interior and exterior design temperatures and building characteristics. Furthermore, the rules used to insure that the user's input for building type are not present in this knowledge base. The user may enter the building type but the value will not be checked.

To complete the heating and cooling loads this knowledge base uses a methodology similar to the one used in the previous knowledge base. A group of rules is first used to derive needed information based on the building type. The derived information is then used as a basis for estimating the heating and cooling loads.

The estimates themselves are based on data presented in the National Mechanical Estimator by Ottaviano (Ottaviano 1987). The data are stored in a sequential data file and are retrieved by the GETADATA2 program. For cooling loads the estimates are based on values of BTU/hr per ft^2 of occupied floor space. For heating loads the BTU/hr per ft^2 factor presented by Ottaviano was modified. Since the data presented by Ottaviano is based on HVAC projects in the New York City area, the heating load BTU/hr per ft^2 factors were assumed to be valid only for the $55^{O}F$ design temperature difference that was calculated for New York City. Thus, when calculating heating loads the equation used is:

HL = SF * HLF * DELTA / 55

where: HL = estimated heating load in BTU/hr
SF = square feet of occupied floor space
HLF = BTU/hr/ft^2 factor from Ottaviano
DELTA = design temperature difference in ^{O}F

The equation for the cooling loads is simply:

CL = SF * CLF

where: CL = estimated cooling load in BTU/hr
SF = square feet of occupied floor space
CLF = BTU/hr/ft^2 factor from Ottaviano

Both the heating and cooling load factors from Ottaviano are tabulated for each of the 47 building types known to the expert system. The GETDATA2 program is capable of retrieving the data based on the selected building type. The rules of the knowledge base are then tasked to properly apply the load.factors.

First the knowledge base determines the value for the heating design temperature difference. Then it calculates the loads depending on the project needs (heating, cooling,

or both) and on the characteristics of the particular building. For example, for the majority of the building the load can be estimated based on one zone only. This fact is identified by rules such as the following:

IF: (1) [BUILDING TYPE] = "BAR_LOUNGE"
THEN: (1) RULE-OF-THUMB LOAD ESTIMATES REQUIRE ONE ZONE ONLY

IF: (1) [BUILDING TYPE] = "HOTEL"
THEN: (1) RULE-OF-THUMB LOAD ESTIMATES REQUIRE MULTIPLE ZONES

In this example, the hotel requires multiple zones because guest rooms, offices, corridors, and conference rooms all have different requirements. The bar, on the other hand, is normally one homogeneous space. To compute the loads for the "single zone" spaces only a few rules are needed:

IF: (1) RULE-OF-THUMB LOAD ESTIMATES REQUIRE ONE ZONE ONLY
and (2) THE PROJECT REQUIRES COOLING or HEATING AND COOLING
THEN: (1) [EST TOTAL C LOAD] = [AREA] * [EST C LOAD FACT]

IF: (1) RULE-OF-THUMB LOAD ESTIMATES REQUIRE ONE ZONE ONLY
and (2) THE PROJECT DOESN'T REQUIRES COOLING or HEATING AND COOLING
THEN: (1) [EST TOTAL C LOAD] = 0

While these rules take care of the cooling loads, a few others are needed for the heating loads. Notice that in the second rule the value of zero has to be provided. If this is not done the knowledge base will ask the user to input the estimated total cooling load anytime that it cannot determine the value by itself.

The process for "multiple zone" buildings is similar to the process described thus far but is more detailed. In hotels and motels, for example, the loads are estimated per guest room, per floor, per average conference room and then a total is compiled. Other examples of multiple zone buildings are office buildings (internal and external zones), apartments (similar to hotels), shopping malls (loads per mall floor and loads per retail

outlet), and department stores (loads per floor). The separation of buildings into these two groups for estimating loads was based on the detail of load information data presented by Ottaviano and the need for detailed data to complete the next function for which this knowledge base is used.

This second function is the selections of a generic system description. A total of eleven choices are used:

1. Unitary through the wall systems
2. Unitary packaged systems
3. Unitary split systems
4. Unitary water source packaged systems
5. Built-up single zone systems
6. Multizone systems
7. Packaged VAV systems
8. Built-up VAV systems
9. Fan-coil unit systems
10. Radiant heating systems
11. Forced air furnaces

The selection rules in this knowledge base are for the most part based on the ASHRAE Handbook 1987 HVAC Systems and Applications (ASHRAE 1987). Thus, the overriding factor on which selection is based is "accepted or common practice." The judgement on just what is common practice is based on the building type, building characteristics and configuration, and on the estimated loads. For example, common practice in efficiency apartments with small loads is to install through-the-wall packaged units, larger apartments in one or two story buildings are served by unitary split systems or roof mounted packaged systems, while high-rise apartment building are served by hydronic fan-coil units or by individual water source heat pumps connected to a boiler/cooling tower loop.

The system selections from this section are converted and sorted by the UPDATE program and the top six selections are sent to the next knowledge base.

Preliminary Cost Knowledge Base

Purpose. The purpose of this knowledge base is to continue the preliminary design work begun by the previous knowledge base. It uses inputs from the two previous knowledge bases and provides, without user inputs in most cases, a preliminary cost, mechanical room space estimate, and practical economic life for each of the systems selected in the last knowledge base.

Programmed Knowledge. The primary inputs for this knowledge base are the systems selected in the preliminary design knowledge base, the building type, the estimated loads, and the project location. This knowledge base uses the GETDATA2 program to access information from three sequential data files. The data found in the files come from several sources.

The simplest function that this knowledge base performs is a data access for the estimated economic life of the selected systems. This value, taken from Ottaviano (1987), is an estimate of the number of years that a system can be expected to operate without requiring replacement. The value is simply provided to the user.

The other two functions that the knowledge base performs are somewhat more complicated and require both data and logic. First is the initial cost estimate. Cost data from both the Ottaviano estimating manual (Ottaviano 1987) and the Means' Mechanical Cost Data 1988 (Mahoney 1987) were compiled for both heating-only and heating and cooling systems. The costs were converted into factors of dollars per ton of refrigeration or dollars per 1000 BTU/hr of heating. The cost for certain systems was split into two different cost factors; one for normal ducted systems and one for non-ducted systems. For example in hotels, hospital patient rooms, and schools fan coil units may be used as free-standing, free-blowing units without ductwork. In other applications, however, the fan-coil units are connected to a complete air distribution system. The rules of the knowledge base use the correct factor (ducted, non-ducted, heating, and heating and cooling) to estimate a preliminary cost for the selected system.

The last function in this knowledge base also required some different methods. The mechanical room space data used by this section was taken from Building Mechanical Systems (Andrews 1977). The rule-of-thumb factors found in this section are diversified and required several steps to incorporate. Certain systems, like multizone systems, can have their required mechanical room space estimated on a ft^2 per ton or ft^2 per 1000 BTU/hr basis. Space for other systems (forced air furnaces and residential split systems, for example) require a factor of ft^2 per each unit. Still others require that space for auxiliary equipment such as pumps be added to the total obtained from a previous calculation.

The following sample rules illustrate one way in which the knowledge base determines the space requirements for a fan-coil unit system:

```
RULE 1
    IF:   (1) [CURRENT SYSTEM] = "Fan-coil unit systems"
  THEN:   (1) [CURRENT SYST SPACE] = [PRELIM SPACE] + 14
          (2) CLEAR([PRELIM SPACE])
  ELSE:   (1) CLEAR(R 1)
```

Note: 14 ft^2 are added to allow space for pumps.

```
RULE 2
    IF:   (1) [CURRENT SYSTEM] = "Fan-coil unit systems"
   and    (2) MECHANICAL ROOM ESTIMATES REQUIRE ONE ZONE
          ONLY
   and    (3) FREE BLOWING SYSTEMS CAN BE USED IN THIS
          BUILDING
   and    (4) THE PROJECT REQUIRES HEATING ONLY
  THEN:   (1) [PRELIM SPACE] = [MECH SPACE FACT] * ([EST
          TOTAL H LOAD] / 1000)
  ELSE:   (1) CLEAR(R 2)

RULE 3
    IF:   (1) THE PROJECT REQUIRES HEATING ONLY
  THEN:   (1) [COST DATA FILE] = "HTGSYST.DAT"

RULE 4
    IF:   (1) [BUILDING TYPE] = "ELEMENTARY_SCHOOL"
  THEN:   (1) MECHANICAL ROOM ESTIMATES REQUIRE ONE ZONE
          ONLY
   and    (2) FREE BLOWING SYSTEMS CAN BE USED IN THIS
          BUILDING
```

RULE 5
IF: (1) [RECOM SYST 3] <> "NONE"
THEN: (1) [CURRENT SYSTEM] = [RECOM SYST 3]
and (2) [SYST 3 SPACE] = [CURRENT SYST SPACE]
and (3) CLEAR([CURRENT SYSTEM])
and (4) CLEAR([CURRENT SYST SPACE])

Assume that the selected building type is "elementary school" and that recommended system three is "Fan-coil unit systems." What the knowledge base is trying to find is a value for system three space. The logic path begins at rule five. Since recommended system three is not equal to "NONE," the then part of rule five assigns the value of the recommended system three variable to the current system variable. Next the rule attempts to assign the value of the current system space variable to the system three space variable (the goal). Not finding a value, the knowledge base begins back chaining and finds that rule one can compute the needed value if the current system is a fan coil unit system. Since this is the case, the rule proceeds but finds the value of the preliminary space variable missing and back-chains in search of that value.

Next the knowledge base finds that rule two can compute the preliminary space. To prove rule two, however, rule four must be looked at to see if all the "if" conditions in rule two are true. Since they are, rule two begins to compute the preliminary space value and has to look up the value of the mechanical space factor in a data file. Before it can do this, however, the knowledge base must know which file to look in. This information is found by chaining to rule three. Once the factor is retrieved, the preliminary and current spaces are computed and the final value is assigned to the system three space variable. Recalling the discussing on recursion, note that several of these rules use "CLEAR" statements to allow variables and rules to be reused. If rule five does not clear the two temporary variables, the next rule that attempts to use these variables will get the same value for current system space that rule five got.

System Selection Knowledge Base

Purpose. This knowledge base provides the user with a rank ordered list of possible HVAC systems for any given project.

Programmed Knowledge. Like the preliminary design knowledge base, this knowledge base makes selections based on "common HVAC practice" as presented in the ASHRAE Handbook 1987 HVAC Systems and Applications (ASHRAE 1987). This knowledge base also bases its decisions on type of building, weather factors, and on the estimated loads from the preliminary design knowledge base. It also looks at other factors such as owner preferences for low first cost, energy efficiency, or flexibility.

The main difference between this knowledge base and the preliminary design knowledge base is the level of detail. While the preliminary design knowledge base selected from eleven possible systems, this knowledge base selects from 55 systems. Recalling that one of the system choices in the previous knowledge base was "Fan-coil unit systems," we notice that now there are six choices for fan-coil unit systems:

1. Ducted two pipe fan coil units.
2. Ducted two pipe fan coil units with electric heat.
3. Ducted four pipe fan coil units.
4. Free discharge two pipe fan coil units.
5. Free discharge two pipe fan coil units with electric heat.
6. Free discharge four pipe fan coil units.

In addition to a fivefold increase in system choices, this knowledge base also uses many more rules to select the systems. It looks more closely at the selection process and bases its decisions on more detail and more rules. For the sake of comparison, note that this knowledge base has about 270 rules and its only function is system selection. The preliminary design knowledge base, on the other hand, has approximately 230 rules to estimate loads and select systems.

Of all the knowledge bases described thus far this comes the closest to fitting the "ideal" expert system function described in the AI literature. It is a pure selection type of knowledge base and does not conduct any type of work that could possibly be completed

by conventional programs. Also, adding a greater level of detail to this knowledge base does not invalidate any of the existing rules, which is an important characteristic of good expert systems.

The results of this knowledge base are passed to both remaining knowledge bases in the expert system. The UPDATE program converts the choices to variables and passes the top eight selections to the equipment and controls selection knowledge bases.

Equipment Selection Knowledge Base

Purpose. The purpose of this knowledge base is to select the major pieces of equipment that make up the systems selected in the previous knowledge base.

Programmed Knowledge. This knowledge base will work on only one or on all of the selected systems. The user indicates which of the systems the knowledge base works with, and he must note the results for each system because the UPDATE program only saves the results of the last system it works with. The UPDATE program could be modified to allow the expert system to save the results for each of these systems in a separate output data file.

The scheme that is used to control the execution of this knowledge base is best illustrated by looking at some of the rules:

Rule 1
IF: (1) THE RECOMMENDED SYSTEM THAT YOU WANT TO WORK WITH IS [[DIST SYST 1]]
THEN: (1) [CURRENT SYSTEM] = [DIST SYST 1]

Rule 2
IF: (1) THE RECOMMENDED SYSTEM THAT YOU WANT TO WORK WITH IS [[DIST SYST 2]]
THEN: (1) [CURRENT SYSTEM] = [DIST SYST 2]

Rule 3
IF: (1) [CURRENT SYSTEM] = Baseboard perimeter radiation
and (2) CENTRAL STEAM IS CURRENTLY AVAILABLE FOR USE IN THIS PROJECT
THEN: (1) Need steam to hot water converter - Probability=90/100

and (2) Need hot water pumps - Probability=100/100
and (3) Need baseboard finned tube convectors - Probability=100/100

First, recall that the UPDATE program passes the top eight results of the previous knowledge base to this knowledge base. These results are stored in the [DIST SYST 1] through [DIST SYST 8] variables. If the previous knowledge base is not used, it is possible for the user to enter system descriptions on his own. The descriptions, however, must be the same as the ones in the system selection knowledge base or the expert system will not recognize them. A simple way of ensuring the correct values are entered is to build a data file using the DATABASE program. Then the GETDATA1 program can be used to insure proper values are entered anytime the user wishes to skip the previous knowledge base. This function can be applied to all knowledge bases to increase the versatility of the expert system, however, it has not been implemented to date.

Once the selected systems are entered (either by the user or by the UPDATE program), eight rules such as rules one and two are activated. The qualifier at the head of these rules has the following text: "THE RECOMMENDED DISTRIBUTION SYSTEM THAT YOU WANT TO WORK ON IS." The value options for this qualifier are the eight variables [DIST SYST 1] through [DIST SYST 8]. Recall that when variables are embedded in qualifiers, the value of the variables is displayed on the input screen. For brevity assume that there are only two [DIST SYST #] variables and that they have the values "Baseboard perimeter radiation" and "Ducted, two pipe fan-coil units." The input screen for the qualifier in question would then look as follows:

THE RECOMMENDED DISTRIBUTION SYSTEM THAT YOU WANT TO WORK ON IS:

1. Baseboard perimeter radiation
2. Ducted two pipe fan-coil units

When the user enters one of the two qualifier variables from the menu, rule one or rule two will be true. The same idea can be extended to up to 30 qualifier values as long as there are as many rules (like rules 1 and 2) as there are qualifier variables.

The tail of the rules, as can be seen, assigns the value of the selected system to a variable called [CURRENT SYSTEM]. The rest of the rules (like rule three) then deal with the current system variable. Thus this scheme allows user selection of a system for the knowledge base to work on.

As rule three indicates, the function of this knowledge base is to determine generic equipment types. There are currently 54 choices in the knowledge base. Like the previous knowledge base this knowledge base can be refined with little or no change to the existing rules. For example, for a higher level of detail rule three above can be modified as follows:

Rule 3

IF:	(1) [CURRENT SYSTEM] = Baseboard perimeter radiation
and	(2) CENTRAL WASTE STEAM IS CURRENTLY AVAILABLE FOR USE IN THIS PROJECT
THEN:	(1) Need shell and tube steam to hot water converter - Probability=85/100
and	(2) Need direct contact heat exchanger - Probability=70/100
and	(3) Need hot water pumps - Probability=100/100
and	(4) Need baseboard finned tube convectors - Probability=100/100

The changes include a refinement in the description of the available steam source and a refinement in the description of the heat exchanger. Since the EXSYS editor automatically tracks these changes the actual mechanics are actually simpler than making the change as done here. Similar changes can be made to parts of the knowledge base while other parts are left unchanged. The importance lies in the level of detail that the programmer wants to achieve.

Controls Selection Knowledge Base

Purpose. The purpose of this knowledge base is to select control schemes for the systems that were selected by the system selection knowledge base.

Programmed Knowledge. The entire discussion presented in the section concerning the equipment selection knowledge base is applicable to this section. The only

difference between these knowledge bases is that one selects equipment and the other control schemes. The schemes selected here are taken from Guide Specifications for HVAC Control Systems (Hittle et al. 1986). The choices in the knowledge base are generic names that are cross-referenced to the guide specification. For example, one choice is "VAV control scheme 1." When this choice is selected, a variable is given the value "VAV control scheme 1 is presented on page A1-3 of controls reference 1," while another variable is given the value "Control's reference 1 is USAF Standardized HVAC Control Systems Technical Specification, (Hittle et al. 1987)." These variables are displayed by EXSYS along with the choices at the end of the consultation. The reason for this awkward procedure is that most control schemes cannot be given short simple names as can systems and equipment. A better procedure could perhaps be developed by using the PROPRINT program to cross-reference the output to the specification. For example, as part of the expert system output the pertinent sections of the specification can be printed as can the design instructions for the selected control scheme. This idea can also be applied to the system and equipment selection knowledge bases where design checklists for each selected system could be generated by the PROPRINT program.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Assessment of Accomplishments

Although the final product that has resulted from this project is far from being a complete expert system, the accomplishments are significant in that collectively they define and demonstrate the concept of an expert system for the design of HVAC systems. The significance of the results presented here is enhanced by the complete absence of similar (HVAC design) expert systems in the literature.

The key accomplishments of this project are:

1. The development of the expert system specification.

2. The development of a prototypical working structure capable of supporting the concepts described in the specification.

3. The identification of HVAC design subject areas suitable for knowledge-based programing.

4. The demonstration of a valid knowledge programming scheme for HVAC design knowledge.

The expert system specification gave direction to the rest of the work. It clearly relates the characteristics of expert systems to the interactive process of HVAC design, and defines all aspects of the expert system. This specification can be used as a guide for further development of this expert system and for the development of other expert systems for HVAC design.

The prototypical structure of the expert system is significant because it implements many of the concepts described in the specification, and most importantly because it does this by adapting a generic expert system shell to the task of HVAC design. The success

of the structure shows that existing expert system technology can be applied to HVAC design. The structure makes good use of the strenghts of the EXSYS shell, and corrects some of its problems.

Along with the structure, the identification of HVAC design subjects suitable for HVAC design was crucial to the success of this project. As has been seen the identification of such knowledge is anything but trivial. The selected knowledge must be of use to the pontential user, and must involve formal reasoning. The survey and the interaction with students were instrumental in providing inputs to the selection of appropriate topics.

Finally the demonstration of a valid knowledge programming scheme is important because it makes this one of the first applications of expert system technology to HVAC design. The programming scheme minimizes user inputs and simplifies the addition of future rules to increase the knowledge of the expert system. Two essential attributes of successful expert systems.

Guidance for Future Work

Shell Selection

Recommended Procedure. Although the EXSYS expert system shell has been very useful for this expert system, a more powerful shell may be needed for future work. Before such a shell is selected it is strongly recommended that the "ideal" expert systems described in the literature be studied (Jackson 1986; Levine et al. 1986; Townsend et al. 1986; Van Horn 1986). After this a readily available shell comparable to EXSYS should be thoroughly studied and compared to the "ideal" expert systems. Familiarity with both the "ideal" expert system concept and with a good shell such as EXSYS will allow for the thorough evaluation of shells being considered for future work.

Since the literature and demonstration programs provided by the vendors and developers of expert system shells do not provide enough detailed information for good comparisons, all evaluations should be based on the actual performance of the shells. To

do this it may be necessary to travel to the developers' places of business as expert system shells are generally sold by the developers and not by local retail outlets. The cost of such trips should be justifiable based on the cost of advanced expert system shells which runs in the range of $1000 to $10,000.

EXSYS Deficiencies. The deficiencies that have been noted in the EXSYS expert system shell should be considered if selecting a replacement for EXSYS during future projects. Briefly, a shell used to replace EXSYS should have the same capabilities as EXSYS plus some or all of the following:

1. Provide a built-in method of allowing the user to enter the level of confidence of his responses to knowledge base questions.

2. Provide a built-in method of using the results of one knowledge base as data in other knowledge bases.

3. Provide frames capabilities that can be used to assign attributes to building types.

4. Provide built-in graphics display capabilities both in the user interface and the report generator. EXSYS has these capabilities but a graphics editor capable of producing ASCII graphics files is needed.

Hardware

The fundamental concept that this expert system be developed for a personal computer does not need to be reconsidered. More than ever the personal computer is the computer of choice for the engineering consulting firms that will most likely be the users of HVAC design expert systems. The hardware problems encountered during this project occurred when the expert system exceeded first the memory limits of the machine used to develop the expert system (512 KB RAM) and second the memory access limits of the MS-DOS operating system (640 KB RAM).

Using a personal computer with more memory and with an operating system capable of addressing the added memory will clearly solve the lack of RAM problem encountered in this project. Current information indicates that the new Microsoft Corporation operating system known as Operating System 2 (MS-OS/2) is capable of

addressing 16 megabytes of RAM on 80286 and 80386 based microcomputers (White 1988). Clearly 16 megabytes of memory is more than adequate to support the development of the expert system to ten orders of magnitude beyond its current size.

As MS-OS/2 compatible expert system shells and programming languages become available the programs developed for this project can be compiled and interfaced with an EXSYS and/or other expert system shell. Currently OS-2 compatible software such as BASIC language compilers and EXSYS are not available.

Knowledge Base Programming

Knowledge-base programming should be the main activity of follow up work to this project. This project has shown that interfacing with an expert system shell is possible in several different ways. It is possible to take a shell like EXSYS and pass data to and from all knowledge bases, it is possible to format the knowledge-base results to obtain meaningful printed outputs and it is possible to allow the shell to directly interface with data access and computation programs to free the user from tedious details. Future work, therefore, should concentrate on expanding the knowledge contained in the knowledge bases as well as correcting inefficiencies in the knowledge bases produced in this project.

The knowledge engineering methods explained by P. W. Brothers (Brothers 1988) should be considered for future work because they provide a thorough, systematic approach to expert system development. However, at least during the early stages of programming, the processes of knowledge acquisition and knowledge programming should not be separated. In other words, the tendency to compile quantities of knowledge before actually attempting to program the knowledge should be avoided. The preferred practice during the early stages is one of acquiring, programming and testing small amounts of knowledge. This approach is a faster route to proficiency in knowledge

programming because the small amounts of knowledge will make errors easier to detect and will better highlight the idiosyncrasies of the expert system shell.

Once proficiency is achieved it may be possible to make better progress by assembling the knowledge, planning the programming and executing the programming. This procedure can be faster because the programmer is well aware of what works and what does not. The danger with using this method too early is that large amounts of work may have to be redone if an unexpected error is found.

Changes to the Current HVAC Design Expert System

Certain specific changes and additions have been identified as necessary for the further development of this program. These changes include additions to the expert system structure and refinements and additions to the knowledge bases.

Addition to Structure. One problem noted in the current structure is that it is not possible for the user to add new building types to the BUILDING.DAT data base. With the DATABASE program the user can safely add locations to the LOCATION.DAT data file as long as the related sequential data files (SUMMER.DAT, WINTER.DAT, and COSTFACT.DAT) are also updated in accordance with the guidance provided in Appendix E of Herrick Laboratory Report HL 89-37 (Camejo and Hittle 1989). However, if the user adds a building type to the BUILDING.DAT data file the expert system will not recognize the newly entered building type and the user will be asked many questions that the expert system would otherwise not need to ask. A solution to this problem is to include rules in the facility analysis knowledge base such that the knowledge base can be used to "learn" about new building types.

The "learning" concept would allow the user to add new building types to the BUILDING.DAT data file. Then the facility analysis knowledge base, faced with a new and unknown building type, would invoke rules used to find relevant information about the unknown building. Of course this learning process requires that the user be familiar

with the building characteristics. Questions about building use, occupancy, construction and required conditions, among others, would identify the building type to the expert system.

Taking this concept further, the expert system could then be programmed to store the newly learned information about the unknown building type in a data file. For the sake of illustrating this concept the data file can be named NEWBLDG.DAT. When the same building type is again entered by a user the expert system can search the NEWBLDG.DAT data file and find the needed information about the previously unknown building type without having to ask for information again. Of course if the building type is new and not yet found in the data file the facility analysis knowledge base will ask the needed questions and save the results for future reference.

The subroutines that exist within the programs of the expert system have the data manipulation capabilities that are needed to implement this "learning" concept. Thus from a programming point of view the idea is feasible. Caution, however, must be taken to keep an inexperienced user from giving erroneous information to the expert system. To do this the "learning" rules in the facility analysis knowledge base must do as much thinking as possible. For example, the expert system must not ask the user to enter whether a facility is a high internal load facility or not. Instead, the expert system should ask questions about the facility, i.e., occupancy and activity levels, appliance types and numbers, etc. and then infer the type of loads that can be expected in the building in question.

Changes in Knowledge Bases. The work conducted during this project uncovered some additional topics considered suitable for inclusion in the knowledge bases of the expert system. These topics were not programmed and are presented in Table 6 as candidates for future work. Adding some or all of these topics is one possible course of action to be taken. Further work can also be done to improve and or expand the knowledge that is already in the expert system.

Table 6

Additional topics for knowledge bases.

KNOWLEDGE BASE	POSSIBLE TOPICS
Facility Analysis	Identify opportunities for heat recovery Adjustment of calculated heating loads -Due to internal heat sources -Due to intermittent operation Identify zoning requirements Identify expected load profiles
Preliminary Design	None noted
Preliminary Cost	Estimated duct space requirements for selected systems Estimate electrical requirements for selected systems Provide energy usage comparison for selected systems
System selection	Refine selection knowledge
Equipment selection	Help with equipment selection from catalogs
Controls selection	Increase number of control schemes

Further work to be done on the knowledge bases is evaluation of the knowledge by HVAC design experts who have not taken part in the development of the expert system. Their inputs could provide a good assessment of the worth of the project and additional ideas for types of knowledge that should be added. Evaluation by students of HVAC design and/or engineers with limited HVAC design experience should also be considered as their point of view will certainly be different than that of recognized experts.

Once the evaluation process is complete, recommendations should be implemented and the evaluation process should be repeated. It is strongly believed that independent evaluation is the key to the advancement of the expert system beyond its current state.

BIBLIOGRAPHY

BIBLIOGRAPHY

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1988 Equipment. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988.

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1972 Fundamentals. New York: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1972.

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1985 Fundamentals. Inch-Pound ed. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1985.

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1987 HVAC Systems and Applications. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1987.

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1986 Refrigeration. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1986.

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Standard: Energy Conservation in New Building Design. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1980.

Andrade, Guillermo, ed. and Degelman, Larry O., ed. A Bibliography of Available Computer Programs in the Area of Heating Ventilating, Air Conditioning, and Refrigeration. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1986.

Andrews, F. T. Building Mechanical Systems. New York: McGraw-Hill Book Company, 1977.

Bradshaw, Vaughn. Building Control Systems. New York: John Wiley and Sons, 1985.

Bratko, Ivan. PROLOG Programming for Artificial Intelligence. Wokingham, England: Addison-Wesley Publishers Limited, 1986.

Brothers, P. W. "Knowledge Engineering for HVAC Expert Systems." In ASHRAE Transactions 1988 Part 1 Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1063-1073.

Camejo, P. J., Hittle, D. C. "An Expert System For The Design of Heating, Ventilating, and Air-Conditioning Systems." Herrick Laboratory Report Purdue University 1989.

Carrier Air-Conditioning Company. Carrier System Design Manual. Part 9, Systems and Applications. Syracuse, N.Y.: Carrier Air-Conditioning Company, 1965.

Crabb, Donald Evan. "80386 Accelerator Boards." Byte, August 1987, 213-216.

Daryanani, Sital. Building Systems Design With Programmable Calculators. New York: McGraw-Hill Book Company, 1980.

EXSYS Rel 3.2.5. EXSYS Incorporated, Albuquerque, N.Mex.

Gero, J. S. "Expert Systems in Design and Analysis." In The National Engineering Conference, Melbourne, March 2-8 1985, by the Institute of Engineering Australia. Barton, Australia: Institute of Engineering Australia, 1985, 211-217.

Haberl, J. S., Smith, L. K., Cooney, K. P., and Stern, F. D. "An Expert System for Building Energy Consumption Analysis: Applications at a University Campus." In ASHRAE Transactions 1988. Part 1, Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1037-1062.

Hittle, Douglas C., Schwenk, David M., and Foran, Kevin J. Design Instructions for HVAC Control Systems. Champaign, Ill.: U. S. Army Construction Engineering Research Laboratory, 1986.

Hittle, Douglas C., Schwenk, David M., and Foran, Kevin J. Guide Specifications for HVAC Control Systems. Champaign, Ill.: U. S. Army Construction Engineering Research Laboratory, 1986.

Jackson, Peter. Introduction to Expert Systems. Wokingham, England: Addison-Wesley Publishing Company, 1986.

Kanzler, Stephen. "INTEL is Providing 80386 Boards to Major Hardware Developers." PC Week, August 19, 1986, 1.

Kapka, Larraine A. and Hudson, John. HVAC Control System Design. Dayton Ohio: Air Force Institute of Technology, 1988.

Kosten, Celal N., ed. and Maher, Mary L., ed. Expert Systems in Civil Engineering. New York: American Society of Civil Engineers, 1986.

Leithauser, David. "Giving Programs That Professional Touch." PC Resource, October, 1987, 74,75.

Levine, Robert I., Drang, Diane E., Edelson, Barry. A Comprehensive Guide to AI and Expert Systems. New York: McGraw-Hill Book Company, 1986.

Liu, S. T., and Kelly, G. E. "Knowledge-Based Front-End Input Generating Program for Building System Simulation." In ASHRAE Transactions 1988, Part 1, Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1074-1084.

Mahoney, William D., ed. Means' Mechanical Cost Data 1988, 11th Annual Edition. Kingston, Mass.: R. S. Means Company, Incorporated., 1987.

Marksjo, B. S. and Hatjiandreou M. "Developing Knowledge-based Systems for Building Design Approval." In The National Engineering Conference, Melbourne, March 2-8 1985, by the Institute of Engineering Australia. Barton, Australia: Institute of Engineering Australia, 1985, 206-210.

Microsoft Quick BASIC Rel. 3.0. Microsoft Corporation, Redmond Wash.

Mueller Associates, Incorporated. Active Solar Thermal Design Manual. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated., 1986.

O'Connel, Colette, ed., Sheviak, Jean K., ed., Browne, Alice R., ed., and Johnson, Samuel V., ed. Directory of Microcomputer Software for Mechanical Engineering Design. New York: Marcel Dekker, Incorporated., 1985.

Ottaviano, Victor B. National Mechanical Estimator by Ottaviano. Melville, New York: Ottaviano Technical Services, Incorporated., 1987.

Simons, G. L. Expert Systems and Micros. Manchester, England: The National Computing Centre Limited, 1985.

Simpson, Henry. Programming The IBM PC User Interface. New York: McGraw-Hill Book Company, 1986.

Soong, Norman L. "The Economics of Scale, Micro-systems and Super Micro-systems for Architects, Contractors, and Engineers." In The IEEE 1985 Microprocessor Forum Conference, by the Institute of Electrical and Electronics Engineers Incorporated. New York: Institute of Electrical and Electronics Engineers Incorporated, 1985, 75-82.

Stanford, Herbert W. III. Analysis and Design of Heating, Ventilating, and Air-Conditioning Systems. With the collaboration of Paykan Safemazadarani. Englewood Cliffs, N.J.: Prentice Hall, 1988.

Summers, Mike, EXSYS Technical Advisor. Interview by author, 27 July 1988. Dayton, Ohio. Telephone conversation.

Townsend, Carl and Feucht, Dennis. Designing and Programming Personal Expert Systems. Blue Ridge Summit, Pa.: Tab Books, Incorporated., 1986.

U. S. Department of Commerce, National Climatic Data Center. Comparative Climatic Data for the United States Through 1985. Asheville, N.C.: U. S. Department of Commerce, National Climatic Data Center, 1986.

U. S. Department of the Air Force. Criteria and Standards for Air Force Construction. Interim Draft ed. January 1986. Washington D.C.: Government Printing Office, 1986.

U. S. Departments of the Air Force, the Army, and the Navy. Facility Design and Planning Engineering Weather Data. Washington D.C.: Government Printing Office, 1985.

Van Horn, Mike. Understanding Expert Systems. Toronto: Bantam Books., 1986.

White, Eva M. "IBM OS/2 Standard Edition." Byte, June 1988, 178-182.

Wilensky, Robert. LISPcraft. New York: W. W. Norton and Company, 1984.

Williams, David A. "Programming with Quick Basic." PC Resource, October, 1987, 54-61.

Winston, Patrick Henry. Artificial Intelligence. 2d ed. Reading, Mass.: Addison-Wesley Publishing Company, 1984.

Wolverton, Van. Supercharging MS DOS. Redmond Wash.: Microsoft Press, 1986.

APPENDICES

Appendix A

Expert System Specification

HVAC DESIGN EXPERT SYSTEM SPECIFICATION

PURDUE UNIVERSITY
RAY W. HERRICK LABORATORIES
WEST LAFAYETTE, INDIANA

REVISION: 1.1
DATE: September 23, 1988

RECORD OF CHANGES

Initial version of this specification completed on March 25, 1987, by Pedro J. Camejo.

Revision 1.0 completed by Pedro J. Camejo on August 30, 1987. Expert system version 1.0 was developed in accordance with this revision of the specification. The revision includes the following changes:

a. Updated bibliography.

b. Updated paragraphs 4.3.1, 4.3.3, 4.4-c, 5.2.2, 5.2.2.4, and 5.2.5.2; deleted paragraphs 5.2.2.2, and 5.2.2.2.4.2; added paragraph 4.4-g. All changes made to reflect the known capabilities of the selected expert system shells which is EXSYS Version 3.2.

c. Completely revised section 15 to match the HVAC design outline found in the August 1987 survey of 120 USAF mechanical engineers conducted by this writer. For survey details see thesis by this writer titled Expert System for the Design of Heating Ventilating and Air-Conditioning Systems. The thesis is scheduled to be submitted to Purdue University in May 1988.

d. Deleted figures.

Revision 1.1 completed by Pedro J. Camejo on September 23, 1988. Expert system version 1.1 was developed in accordance with this revision of the specification. The revision includes the following changes:

a. Updated bibliography.

SECTION 1

GENERAL

1. BIBLIOGRAPHY: The following bibliography was used in the development of this specification and/or in the development of expert system versions developed in accordance with this specification:

1.1 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1985 Fundamentals. Inch-Pound ed. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1985.

1.2 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. ASHRAE Handbook 1987 HVAC Systems and Applications. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1987.

1.3 Andrade, Guillermo, ed. and Degelman, Larry O., ed. A Bibliography of Available Computer Programs in the Area of Heating Ventilating, Air Conditioning, and Refrigeration. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1986.

1.4 Andrews, F. T. Building Mechanical Systems. New York: McGraw-Hill Book Company, 1977.

1.5 Bradshaw, Vaughn. Building Control Systems. New York: John Wiley and Sons, 1985.

1.6 Bratko, Ivan. PROLOG Programming for Artificial Intelligence. Wokingham, England: Addison-Wesley Publishers Limited, 1986.

1.7 Brothers P. W. "Knowledge Engineering for HVAC Expert Systems." In ASHRAE Transactions 1988 Part 1 Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1063-1073.

1.8 Carrier Air-Conditioning Company. Carrier System Design Manual. Part 9, Systems and Applications. Syracuse, N.Y.: Carrier Air-Conditioning Company, 1965.

1.9 Crabb, Donald Evan. "80386 Accelerator Boards." Byte, August 1987, 213-216.

1.10 Daryanani, Sital. Building Systems Design With Programmable Calculators. New York: McGraw-Hill Book Company, 1980.

1.11 EXSYS Rel 3.2.5. EXSYS Incorporated, Albuquerque, N.Mex.

1.12 Gero, J. S. "Expert Systems in Design and Analysis." In The National Engineering Conference, Melbourne, March 2-8 1985, by the Institute of Engineering Australia. Barton, Australia: Institute of Engineering Australia, 1985, 211-217.

1.13 Haberl, J. S., Smith, L. K., Cooney, K. P., and Stern, F. D. "An Expert System for Building Energy Consumption Analysis: Applications at a University Campus." In ASHRAE Transactions 1988. Part 1, Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1037-1062.

1.14 Hittle, Douglas C., Schwenk, David M., and Foran, Kevin J. Design Instructions for HVAC Control Systems. Champaign, Ill.: U. S. Army Construction Engineering Research Laboratory, 1986.

1.15 Hittle, Douglas C., Schwenk, David M., and Foran, Kevin J. Guide Specifications for HVAC Control Systems. Champaign, Ill.: U. S. Army Construction Engineering Research Laboratory, 1986.

1.16 Jackson, Peter. Introduction to Expert Systems. Wokingham, England: Addison-Wesley Publishing Company, 1986.

1.17 Kanzler, Stephen. "INTEL is Providing 80386 Boards to Major Hardware Developers." PC Week, August 19, 1986, 1.

1.18 Kosten, Celal N., ed. and Maher, Mary L., ed. Expert Systems in Civil Engineering. New York: American Society of Civil Engineers, 1986.

1.19 Leithauser, David. "Giving Programs That Professional Touch." PC Resource, October, 1987, 74,75.

1.20 Levine, Robert I., Drang, Diane E., Edelson, Barry. A Comprehensive Guide to AI and Expert Systems. New York: McGraw-Hill Book Company, 1986.

1.21 Liu, S. T., and Kelly, G. E. "Knowledge-Based Front-End Input Generating Program for Building System Simulation." In ASHRAE Transactions 1988. Part 1, Technical and Symposium Papers Presented at the 1988 Winter Meeting In Dallas, by the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated., Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated, 1988, 1037-1062.

1.22 Mahoney, William D., ed. Means' Mechanical Cost Data 1988, 11th Annual Edition. Kingston, Mass.: R. S. Means Company, Incorporated., 1987.

1.23 Marksjo, B. S. and Hatjiandreou M. "Developing Knowledge-based Systems for Building Design Approval." In The National Engineering Conference, Melbourne, March 2-8 1985, by the Institute of Engineering Australia. Barton, Australia: Institute of Engineering Australia, 1985, 206-210.

1.24 Microsoft Quick BASIC Rel. 3.0. Microsoft Corporation, Redmond Wash.

1.25 Mueller Associates, Incorporated. Active Solar Thermal Design Manual. Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Incorporated., 1986.

1.26 O'Connel, Colette, ed., Sheviak, Jean K., ed., Browne, Alice R., ed., and Johnson, Samuel V., ed. Directory of Microcomputer Software for Mechanical Engineering Design. New York: Marcel Dekker, Incorporated., 1985.

1.27 Ottaviano, Victor B. National Mechanical Estimator by Ottaviano. Melville, New York: Ottaviano Technical Services, Incorporated., 1987.

1.28 Simons G. L. Expert Systems and Micros. Manchester, England: The National Computing Centre Limited, 1985.

1.29 Simpson, Henry. Programming The IBM PC User Interface. New York: McGraw-Hill Book Company, 1986.

1.30 Soong, Norman L. "The Economics of Scale, Micro-systems and Super Micro-systems for Architects, Contractors, and Engineers." In The IEEE 1985 Microprocessor Forum Conference, by the Institute of Electrical and Electronics Engineers Incorporated. New York: Institute of Electrical and Electronics Engineers Incorporated, 1985, 75-82.

1.31 Stanford, Herbert W. III. Analysis and Design of Heating, Ventilating, and Air-Conditioning Systems. With the collaboration of Paykan Safemazadarani. Englewood Cliffs, N.J.: Prentice Hall, 1988.

1.32 Townsend, Carl and Feucht, Dennis. Designing and Programming Personal Expert Systems. Blue Ridge Summit, Pa.: Tab Books, Incorporated., 1986.

1.33 U. S. Department of Commerce, National Climatic Data Center. Comparative Climatic Data for the United States Through 1985. Asheville, N.C.: U. S. Department of Commerce, National Climatic Data Center, 1986.

1.34 U. S. Departments of the Air Force, the Army, and the Navy. Facility Design and Planning Engineering Weather Data. Washington D.C.: Government Printing Office, 1985.

1.35 Van Horn, Mike. Understanding Expert Systems. Toronto: Bantam Books., 1986.

1.36 Wilensky, Robert. LISPcraft. New York: W. W. Norton and Company, 1984.

1.37 Williams, David A. "Programming with Quick Basic." PC Resource, October, 1987, 54-61.

1.38 Winston, Patrick Henry. Artificial Intelligence, 2d ed. Reading, Mass.: Addison-Wesley Publishing Company, 1984.

1.39 Wolverton, Van. Supercharging MS DOS. Redmond Wash.: Microsoft Press, 1986.

2. SPECIFICATION:

2.1 Objective: The objective of this specification is to provide a complete and concise guide for the development of an expert system for the design of heating, ventilating and air-conditioning (HVAC) systems. It is understood that the development of such an expert system is evolutionary in nature, and thus no initial document can specify all of the requirements of such a system. It is the intent, therefore, that this specification shall be revised periodically such that it will always satisfy the following goals:

a. Provide continuity between researchers such that the directions of previous researchers are understood by subsequent researchers. A related goal is to provide a history of the expert system's development.

b. Provide the current researchers with an up-to-date specification to guide their system development efforts.

c. Provide the current researchers with a ready reference that explains their work.

The revisions will naturally lead to a tightening of this specification. Where the initial specification will be general, subsequent revisions will be more specific.

2.1.1 Support software programs: Conventional HVAC design software will be used in support of this expert system. It is not the objective of this specification to guide any development of support software beyond specifying the required software and the interface between such software and the expert system.

2.2 Revisions: The revision of this specification shall be as follows:

2.2.1 The specification shall be revised whenever major changes occur to the artificial intelligence (AI) programming aspects of the expert system. Examples of such changes are as follow:

a. Change in computer hardware such as IBM-PC to Apple Macintosh.

b. Change in programming language or version of programming language.

c. Change in expert system shell.

d. Modification to the expert system shell such as change in inference method, user interface, and interface with support software programs.

2.2.2 The specification shall be revised whenever major changes occur to the knowledge base of the expert system. Examples of such changes are as follow:

a. Change in support software or change in version of support software.

b. Addition or deletion of capabilities such as adding knowledge for the design of digital control systems, or the design of commercial kitchen ventilation.

2.2.3 All Changes beyond minor grammatical corrections shall be documented.

2.2.3.1 The title page of this specification shall reflect the revision number and the date of the revision.

2.2.3.2 Revision numbers shall begin with 1.0 and shall increase by tenths (1.0, 1.1, 1.2,...1.9, 2.0, ...).

2.2.3.3 The "changes" section located directly after the table of contents shall contain a brief (one paragraph) history of each specification revision. The following statement will be included when applicable: "Expert system versions ___,___,and___ were developed in accordance with this revision of the specifications." See paragraph 3.3 for the expert system version numbering scheme.

3. EXPERT SYSTEM GENERAL

3.1 <u>Objective:</u> The objective of this research is to develop an expert system for the design of heating, ventilating, and air-conditioning systems, and to test this system through its application on several designs.

3.2 <u>Scope:</u> The expert system, once developed, shall have the reasoning capabilities, expert knowledge, and flexibility specified in this document. Each requirement of the expert system was conceptualized to satisfy the known and or perceived needs of the intended users.

3.3 <u>Revisions:</u> Each version of the expert system shall be numbered with the revision number of the specification used to develop that version of the expert system. The expert system number shall have an added dash and an added third digit to allow for different versions of the expert

system that are developed under the same specification. For example the first, second and third expert system versions to be derived from specification revision number 1.2 shall be numbered 1.2-1, 1.2-2, and 1.2-3 respectively.

4. EXPERT SYSTEM PROGRAMMING CONSIDERATIONS:

4.1 Programming Tools: The expert system shall be developed using the EXSYS expert system shell and the Quick Basic programming language.

4.2 Development Continuity: Regardless of the tool used to develop the expert system (AI languages or expert system shells), the expert system shall contain detailed comments and shall be written using accepted programming practices such as proper indentation of program lines for readability (pp function in LISP), and use of small subroutines for ease of debugging and revision. The intent is to make the program as self explanatory as possible such that future researchers will have minimum trouble in continuing development of the system.

4.3 Separation of Functional Area: Regardless of the method used to develop the expert system, the expert system shall be made up of four distinct functional areas.

4.3.1 *Inference Engine: The inference* engine shall be capable of backward chaining or both forward and backwards chaining search procedures.

4.3.2 Knowledge Data Base: It shall be possible to easily update the knowledge data base without affecting the inference engine.

4.3.3 Knowledge Data Base Editor: The data base editor shall be capable of being disabled to prevent the end user from using the editor to alter some of the system's knowledge data bases. Some knowledge bases shall be left open to editing by the end user. This is so the end user can change, for example, the expert systems knowledge of the conventional software at its disposal. See paragraph 5.2.2.4.

4.3.4 User Interface: The user interface is specified in paragraph 5.2.3.

4.4 Programming Tool Selection Criteria: The following criteria will be used to select the expert system programming tool:

a. Compatibility with specified computer hardware

b. Separation of functional areas as specified in paragraph 4.3.

c. Backward chaining inference engine.

d. Menu driven input and natural language output.

e. Cost.

f. Ease of use (development).

g. Capability to interface with conventional software programs.

5. EXPERT SYSTEM USE CONSIDERATIONS

5.1 General: The primary intended users of this expert system are mechanical engineers with limited HVAC design experience, and graduates of mechanical engineering or HVAC technology curriculums (also with limited HVAC design experience). Additional users that may find this expert system very useful are HVAC maintenance technicians, contractors (both general and mechanical), architects, electrical engineers, construction inspectors, and mechanical engineers with HVAC design experience. It is not intended that this expert system be powerful enough to satisfy all of the needs of all of the possible users. The intent is to satisfy most of the HVAC design needs of the novice mechanical engineer. In doing this it is expected that the expert system will have features that are to some degree useful to all of the possible users listed above.

5.2 Features: The user oriented features of the expert system shall be as follow:

5.2.1 The expert system shall run on a microcomputer. Architect and engineering (A&E) firms, military construction engineering offices, consulting engineers, and HVAC contracting firms are targeted as the primary users of this expert system. These users are more likely to be able to access an expert system using a microcomputer because they are reluctant to spend money on main frame service bureau computing. They will also be reluctant to buy expensive single purpose computer hardware.

5.2.2 The expert system shall interface with conventional HVAC design software programs. A search through the literature [1.3, 1.10, 1.25, and 1.26] shows an abundance of conventional HVAC design software, both public domain and proprietary. The expert system shall have full knowledge of the specific conventional software at the disposal of the user and thus at the disposal of the expert

system. The expert system shall interface with the software available to it.

5.2.2.1 The expert system shall "operate" the supporting software directly as the need arises. The expert system shall query the user for the information needed to run the required software, then the expert system will pass the input data to the supporting software and make use of the output without further help from the user.

5.2.2.2 Paragraph deleted.

5.2.2.3 The expert system shall have the capability of performing simple calculations on its own without having to call on supporting programs. Areas given dimensions and shape, flow given fluid velocity and geometry of conduit, and friction loss given loss per unit length are some of the calculations that the expert system shall perform The intent is not to duplicate available software in the expert system but to provide the expert system with the common place calculations that it will need to do its work.

5.2.2.4 The expert system shall have the capability of allowing any HVAC design expert to initiate the expert system to the supporting software that is available to it. In other words, the expert system must be capable of having its knowledge of its available supporting programs replaced by knowledge of new supporting programs. This must be a function that is available to the users. The intent is to make the expert system independent of proprietary programs. Like a human expert, the expert system must be capable of working at any design office using the HVAC design software that is available at that office. Like any human expert the system must be taught how to use a particular program. This teaching function shall be accomplished by leaving applicable knowledge data bases open to editing by the end user as specified in paragraph 4.3.3.

5.2.2.4.1 To insure that an improper program is not used the expert system shall ask questions whenever its supporting program knowledge is being revised. The questions shall allow the expert system to determine if the proposed change is correct. The system shall allow the knowledge change if the intended change is correct and it shall refuse the change if the intended change is not correct. The intent, for example, is to keep an inexperienced user from replacing the systems knowledge of a water piping design program with a refrigerant piping design program. Once the new support program knowledge is put into the expert system, the expert system shall be able to work with the new program as specified above.

5.2.2.4.2 Paragraph deleted.

5.2.3 The expert system shall be user friendly.

5.2.3.1 The expert system shall ask questions in natural language and shall answer questions in natural language. Because of the complexity of natural language inputs, the expert system shall have a menu-driven input.

5.2.3.2 The expert system shall be capable of guiding the user through the use of the expert system.

5.2.3.3 Input errors by the user shall cause the expert system to return to its previous non-error condition, inform the user that an error in input has occurred and instruct the user on how to continue with his intended work.

5.2.3.4 The expert system shall be capable of proceeding with its work even if the user does not have all of the inputs that the expert system would like to have. When the user does not have the desired inputs the expert system shall accompany its answers with a warning that due to lack of vital information the current answer is of questionable certainty.

5.2.4 The expert system shall be capable of providing simple answers to simple questions. For example asking the system to determine the percent outside air in a mixed air stream shall not require a detailed psychrometric analysis.

5.2.5 The expert system shall be capable of addressing the different phases of system design, and construction individually of each other to allow the user to interact with owners, architects, other engineers, builders, inspectors, and facility users.

5.2.5.1 Regardless of the level used, the expert system's printed output shall be a professional quality document, formatted to present the information in a logical and easy to reference manner.

5.2.5.2 The system shall have a rule-of-thumb database for fast designs and cost estimates. The rule-of-thumb database shall be open for user update (see paragraph 4.3.3). This database shall allow the user to provide preliminary budget figures of cost, mechanical room size, overall mechanical equipment space requirements, cooling and heating loads and other project conceptualization data at a minimum cost in time.

5.2.5.3 The expert system shall have the capability of completing a detailed feasibility study. This feature shall develop a multi system proposal complete with life cycle costs and system descriptions.

5.2.5.4 The expert system shall have the capability of completing a detailed design complete with life cycle costing, detailed calculations (using support programs, detailing information (such as possible problem areas to be on the lookout for, and production drawings if a graphics support program is available to the user.

5.2.5.5 The expert system shall be capable of accessing a database such that a question to the expert system about a particular HVAC component will be answered by the expert system by way of a through description of how the HVAC component should be properly used. The intent is to aid inexperienced construction inspectors in discovering faulty installation practices.

5.2.6 The expert system shall be capable of alerting the novice designer of possible trouble areas in any given design.

5.2.6.1 Regardless of the level of detail of the work done by the expert system (rule-of-thumb design, detailed design, simple questions etc.) the system shall provide a certainty factor with each answer. The certainty factor shall range from 0.0 to 1.0 where 1.0 implies that the answer is a proven fact and 0.0 implies that the expert system was not able to determine an answer.

5.2.6.2 In addition to the certainty factor the program shall provide, when applicable, a rule of thumb check to its calculations and decisions. The inputs used for any particular calculation or decision shall be supplied along with the results.

5.2.6.3 For every HVAC system that the expert system recommends and designs, the expert system shall provide the user with a list of potential problem areas that the user should consider in his production drawings and specifications. The intent here is to cover the minor details that can mean the difference between a good design and a poor one. Examples of such details would be freeze protection, maintenance considerations, and condenser air recirculation.

SECTION 15

HVAC DESIGN KNOWLEDGE DATABASE

1. OBJECTIVE: The objective of this section of this specification is to provide the guidance for the development of the HVAC design knowledge database of this expert system.

2. HVAC DESIGN PRODUCTION SCHEDULE:

2.1 Preliminary Design: [1.2, 1.25] The preliminary design steps are as follow:

2.1.1 Consult the facility owner(s), user(s), and the other members of the design team to obtain as much as possible of the following data:

a. Budget figures. First cost and operating costs.

b. Required completion date of preliminary design.

c. All available plans and project descriptions such as use, occupancy, future expansion plans, possibility of future space realocation, and type of construction.

d. Owner/user equipment and or system preferences to include specific trade names.

e. Required performance, temperature and humidity control, noise criteria, air quality criteria etc.

f. Available fuels, cooling water, existing central refrigeration/heating plants.

2.1.2 In the case of a retrofit project obtain data through site visits.

2.1.3 Prioritize the performance, initial cost, operating costs noise criteria etc. as perceived by the owner(s), user(s), and other members of the design team.

2.1.4 Make preliminary calculations substituting assumed values for data not yet available.

2.1.5 Select possible systems based on the application and on the system's ability to match the prioritize list of design requirements.

2.1.6 Complete a single line layout of each system that can meet the project design criteria and calculate

equipment space requirements. In the case of retrofit projects note the systems that are too large for the available equipment space.

2.1.7 Complete life cycle cost for systems that can meet the project design criteria. Indicate certainty of cost figures with plus or minus percent of total estimate.

2.1.8 Complete preliminary design submittal package to include the following for each system:

a. Life cycle cost figures.

b. Projected system reliability relative to other proposed systems.

c. Projected system performance, ability of system to meet the design criteria as compared to the other proposed systems.

d. System space requirements and system layout.

e. Boiler plate description of system to include operation and maintenance requirements and a list of manufacturers of this type of system.

2.1.9 In addition to the above list the submittal should include:

a. The data used to generate the preliminary design to include data provided by owner as well as data assumed by the engineer or collected at site.

b. Table comparing all the proposed systems.

c. Recommended system and reasons for recommendation.

2.1.10 After the preliminary design is reviewed by all concerned and a final system decided upon proceed to design phase.

2.2 <u>Design:</u> [1.2, 1.25] The design phase parallels the preliminary design phase with a large increase in detail and accuracy as follows:

2.2.1 Insure that all data used for the preliminary design is still accurate and obtain accurate values for data that was not available during preliminary design.

2.2.2 Reacomplish calculations for the chosen system.

2.2.3 Layout and design the air distribution.

2.2.4 Layout and design the piping systems.

2.2.5 Select equipment from catalogs and insure that the equipment space is available.

2.2.6 Complete a detailed estimate and insure that the costs are within budget.

2.2.7 Complete production drawings and specifications. Submit the design for final review.

2.2.8 During the design process coordinate with other members of the design team and make submittals of partial design at specified design-percent-complete intervals.

2.3 <u>Post Design:</u>

2.3.1 Complete plan and specification addenda.

2.3.2 Checking of Shop Drawings and Submittals.

2.3.3 Construction Field Inspections.

3. <u>HVAC DESIGN KNOWLEDGE:</u>

3.1 The following HVAC systems shall be incorporated into the knowledge base of the expert system:

3.1.1 Heating:

a. Gas/oil fired furnace.

b. Gas fired radiant heater.

c. Unit Heater.

d. Gas/oil fired hot water systems.

e. Electric resistance heating.

f. Heat pump.

3.1.2 Refrigeration:

a. Reciprocating compressor.

b. Centrifugal compressor.

c. Absorption.

3.1.3 Heat Rejection:

a. Air cooled condenser.

b. Water cooled condenser.

c. Evaporative condenser.

d. Cooling tower.

3.1.4 Cooling.

a. Packaged direct expansion.

b. Packaged gas fired absorption.

c. Heat pump.

d. Direct expansion (built up).

e. Chilled water.

f. Evaporative cooler.

3.1.5 Piping Systems:

a. Four pipe.

b. Three pipe.

c. Two pipe.

3.1.6 Air Supply and Distribution:

a. Single zone.

b. Single duct constant volume with reheat.

c. Single duct VAV.

d. Dual duct VAV.

e. Multi zone.

f. Fan coil units.

3.1.7 Ventilation Systems:

a. Commercial kitchen exhaust.

b. Industrial exhaust (particulates, fumes etc.).

3.1.8 Scope of Systems:

a. Fractional to 20 Ton Refrigeration (TR) systems.

b. 20 to 100 TR systems.

c. Greater than 100 TR systems.

3.1.9 Scope of Buildings:

a. Less than 5,000 Sq Ft.

b. 5,000 - 10,000 Sq Ft.

c. 10,000 - 50,000 Sq Ft.

d. Single story.

e. Two story.

f. Three to Four story.

g. Five plus story.

3.2 The following HVAC design topics shall be incorporated into the knowledge base of the expert system:

a. Cost estimating.

b. Energy use estimating.

c. Economic analysis and life cycle costing.

d. Specifications.

e. Construction details (code requirements equipment supports, safety equipment, system constructibility etc.).

f. Maintenance details (space and access requirements, test ports and test instruments, maintenance accessories such as floor drains, lights, ladders etc.).

g. Psychrometrics.

h. Ventilation/infiltration calculations.

i. Load calculations.

j. System selection (matching the proper system to the given application for comfort, reliability and efficiency) (are you current with the HVAC industry?).

k. Equipment selection (matching components to each other for system reliability and energy efficiency).

l. Equipment noise control.

m. Air distribution noise control.

n. Air distribution design (temperature, humidity, air motion, radiant temperature, odors and particulates).

o. Duct design and fan selection.

p. Piping design and pump selection.

q. Controls system design (are you current with industry changes?).

3.3 The following type of conventional HVAC design programs shall be incorporated into the knowledge base of the expert system:

a. Heating and cooling load program

b. U factor calculation program

c. Duct sizing program

d. Water pipe sizing program

e. Refrigerant pipe sizing program

f. Energy consumption estimation program

g. Construction cost estimating program

h. Maintenance cost estimating program

i. Life cycle cost estimating program

Appendix B

Survey Questionnaire

SURVEY FOR HVAC DESIGN EXPERT SYSTEM

INSTRUCTIONS:

(1) Participation in this survey is voluntary. There will be no adverse consequences to individuals who elect not to participate.

(2) DO NOT enter your name or social security number or otherwise identify yourself on the answer sheet.

(3) Use a #2 pencil to answer questions on the answer sheet provided. As the survey will be machine scored, be sure to completely darken the circle corresponding to your answer for each question.

(4) Enter only one answer per question.

(5) Follow the instructions before each section of the questionnaire. Take care to match the question numbers with the numbers on the answer sheet.

(6) Return the answer sheet only, in the envelope provided.

****

SECTION I

****

INSTRUCTIONS: Answer questions 1 to 3. These questions are self explanatory.

1. What is your degree?

   (a) BSME
   (b) MSME
   (c) Mechanical Engineering Technology Degree
   (d) Other (non-mechanical engineering degree)

2. Have you completed any HVAC design courses such as the courses taught at the AFIT School of Civil Engineering or did you include any HVAC courses in your college studies?

   (a) Yes
   (b) No

3. What is your total HVAC design or design review experience?

(a) I have never worked in HVAC design. <u>(DO NOT complete SECTION II of this questionnaire, proceed to SECTION III)</u>

(b) Less than four years HVAC design experience.

(c) Four or more years HVAC design experience.

****

SECTION II

****

<u>INSTRUCTIONS:</u> Rate the following HVAC systems based on how often you have designed/design reviewed each type of system in the course of your HVAC design experience. Pick the statement a through c which most closely matches your work experience with each type of system.

(a) I have often worked on the design/design review of this type of system.

(b) I have sometimes worked on the design/design review of this type of system.

(c) I have never worked on the design/design review of this type of system.

<u>HEATING</u>

4. (a) (b) (c) Gas/oil fired furnace.

5. (a) (b) (c) Gas fired radiant heater.

6. (a) (b) (c) Unit Heater.

7. (a) (b) (c) Gas/oil fired hot water systems.

8. (a) (b) (c) Electric resistance heating.

9. (a) (b) (c) Heat pump.

<u>REFRIGERATION</u>

10. (a) (b) (c) Reciprocating compressor.

11. (a) (b) (c) Centrifugal compressor.

12. (a) (b) (c) Absorption.

HEAT REJECTION

13. (a) (b) (c) Air cooled condenser.

14. (a) (b) (c) Water cooled condenser.

15. (a) (b) (c) Evaporative condenser.

16. (a) (b) (c) Cooling tower.

COOLING

17. (a) (b) (c) Packaged direct expansion.

18. (a) (b) (c) Packaged gas fired absorption.

19. (a) (b) (c) Heat pump.

20. (a) (b) (c) Direct expansion (built up).

21. (a) (b) (c) Chilled water.

22. (a) (b) (c) Evaporative cooler.

PIPING SYSTEMS

23. (a) (b) (c) Four pipe.

24. (a) (b) (c) Three pipe.

25. (a) (b) (c) Two pipe.

AIR SUPPLY AND DISTRIBUTION

26. (a) (b) (c) Single zone.

27. (a) (b) (c) Single duct constant volume with reheat.

28. (a) (b) (c) Single duct VAV.

29. (a) (b) (c) Dual duct VAV.

30. (a) (b) (c) Multi zone.

31. (a) (b) (c) Fan coil units.

VENTILATION SYSTEMS

32. (a) (b) (c) Commercial kitchen exhaust.

33. (a) (b) (c) Industrial exhaust (particulates, fumes etc.).

SCOPE OF SYSTEMS

34. (a) (b) (c) Fractional to 20 Ton Refrigeration (TR) systems.

35. (a) (b) (c) 20 to 100 TR systems.

36. (a) (b) (c) Greater than 100 TR systems.

SCOPE OF BUILDINGS

37. (a) (b) (c) Less than 5,000 Sq Ft.

38. (a) (b) (c) 5,000 - 10,000 Sq Ft.

39. (a) (b) (c) 10,000 - 50,000 Sq Ft.

40. (a) (b) (c) Single story.

41. (a) (b) (c) Two story.

42. (a) (b) (c) Three to Four story.

43. (a) (b) (c) Five plus story.

****

SECTION III

****

INSTRUCTIONS: Rate the following HVAC design topics based on the extent of your knowledge about each topic. Pick the statement **a** through **d** which most closely matches your knowledge about each topic.

(a) I am very knowledgeable about this topic and can apply my knowledge to design work with the aid of reference data.

(b) I am knowledgeable about this topic and can apply my knowledge to design work if I first review my old textbooks and notes.

(c) I am familiar with this topic but I cannot use it for design work without the help of more experienced engineers.

(d) I am not familiar with this topic.

44. (a) (b) (c) (d) Cost estimating.

45. (a) (b) (c) (d) Energy use estimating.

46. (a) (b) (c) (d) Economic analysis and life cycle costing.

47. (a) (b) (c) (d) Specifications.

48. (a) (b) (c) (d) Construction details (code requirements equipment supports, safety equipment, system constructibility etc.).

49. (a) (b) (c) (d) Maintenance details (space and access requirements, test ports and test instruments, maintenance accessories such as floor drains, lights, ladders etc.).

50. (a) (b) (c) (d) Psychrometrics.

51. (a) (b) (c) (d) Ventilation/infiltration calculations.

52. (a) (b) (c) (d) Load calculations.

53. (a) (b) (c) (d) System selection (matching the proper system to the given application for comfort, reliability and efficiency) (are you current with the HVAC industry?).

54. (a) (b) (c) (d) Equipment selection (matching components to each other for system reliability and energy efficiency).

55. (a) (b) (c) (d) Equipment noise control.

56. (a) (b) (c) (d) Air distribution noise control.

57. (a) (b) (c) (d) Air distribution design (temperature, humidity, air motion, radiant temperature, odors and particulates).

58. (a) (b) (c) (d) Duct design and fan selection.

59. (a) (b) (c) (d) Piping design and pump selection.

60. (a) (b) (c) (d) Controls system design (are you current with industry changes?).

Appendix C

Survey Results

Table C1

Overall results of survey for HVAC design expert system.

SECTION I: Characteristics of respondents all of whom are active duty USAF Civil Engineering Officers. Tabulated numbers are percent of a total 92 respondents.

1. DEGREE	97% BSME	2% MSME	1% BSME TECH.
2. FORMAL HVAC DESIGN TRAINING	84% YES	16% NO	
3. HVAC DESIGN EXPERIENCE	15% NONE	74% < 4 YRS	11% > 4 YRS

SECTION II: Outline of HVAC design experience by system types. Note that the 15% respondents who in question three above indicated that they had no HVAC design experience were instructed not to respond to this section. Thus the tabulated values are percents of 78 respondents with HVAC design experience.

"A" FREQUENTLY USED <---to---> "C" INFREQUENTLY USED

No.	SYSTEM DESCRIPTION	A	B	C
	HEATING			
4	Gas/oil furnaces	16%	51%	32%
5	Gas radiant heaters	8%	31%	59%
6	Unit heaters	24%	60%	14%
7	Gas/oil fired boilers	23%	56%	19%
8	Elect. resistance heat	4%	61%	32%
9	Heat pumps	12%	49%	39%
	REFRIGERATION			
10	Reciprocating	35%	50%	15%
11	Centrifugal	7%	50%	42%
12	Absorption	0%	15%	83%

Table C1 Continued.

HEAT REJECTION				
13	Air Cooled	50%	41%	9%
14	Water cooled	18%	46%	36%
15	Evaporative	7%	40%	52%
16	Cooling tower	14%	58%	28%
COOLING				
17	Packaged DX	42%	47%	10%
18	Packaged absorption	1%	7%	91%
19	Heat pump	13%	44%	42%
20	Built-up DX	16%	50%	34%
21	Chilled water	45%	48%	7%
22	Evaporative Cooler	14%	32%	54%
PIPING SYSTEMS				
23	Four pipe	13%	50%	37%
24	Three pipe	7%	36%	57%
25	Two Pipe	41%	53%	6%
AIR SUPPLY/DISTRIBUTION				
26	Single zone	64%	32%	4%
27	Sgl. duct w/reheat	20%	44%	36%
28	Single duct VAV	12%	47%	41%
29	Dual duct VAV	2%	18%	80%
30	Multi zone	39%	45%	16%
31	Fan coil units	37%	54%	9%
VENTILATION SYSTEMS				
32	Commercial Kitchen exh.	11%	54%	35%

Table C1 Continued.

33	Industrial exhaust	12%	54%	34%
SCOPE OF SYSTEMS				
34	Fractional to 20 TR	54%	46%	0%
35	20 to 100 TR	22%	62%	16%
36	Greater than 100 TR	5%	26%	69%
SCOPE OF BUILDINGS				
37	Less than 5,000 SF	50%	42%	8%
38	5,000 to 10,000 SF	37%	57%	6%
39	10,000 to 50,000 SF	18%	68%	14%
40	Single story	62%	36%	2%
41	Two story	16%	56%	28%
42	Three to four story	6%	26%	68%
43	Five plus story	1%	8%	91%

SECTION III: Self perceived knowledge of HVAC design topics an "A" reponse indicates very knowledgeable about a given topic while a "D" response indicates lack of familiarity with a given topic. The tabulated numbers are percents of a total 92 respondents.

"A" KNOWLEDGEABLE <-----to-----> "D" NOT KNOWLEDGEABLE

No.	DESCRIPTION OF TOPIC	A	B	C	D
44	Cost estimating	55%	28%	14%	3%
45	Energy use estimating	32%	50%	13%	5%
46	Life cycle costing	17%	49%	27%	7%
47	Specifications	42%	38%	14%	6%
48	Construction details	11%	36%	40%	13%
49	Maintenance details	19%	32%	33%	16%

Table C1 Continued.

50	Psychrometrics	44%	46%	6%	4%
51	Vent./infiltration	42%	48%	6%	4%
52	Heat/Cool load calc.	47%	42%	10%	1%
53	System selection	22%	38%	30%	10%
54	Equipment selection	21%	34%	32%	13%
55	Equipment noise control	4%	28%	42%	26%
56	Air dist. noise control	18%	35%	33%	14%
57	Air dist. design	27%	43%	19%	11%
58	Duct design/fan select.	34%	47%	12%	7%
59	Piping dsgn./pump select.	34%	41%	20%	5%
60	Control system design	16%	28%	29%	27%